

Zhihan Lyu *Editor*

Applications of Generative AI

Applications of Generative AI

Zhihan Lyu

Editor

Applications of Generative AI



Springer

Editor

Zhihan Lyu
Department of Game Design
Uppsala Universitet
Visby, Sweden

ISBN 978-3-031-46237-5

ISBN 978-3-031-46238-2 (eBook)

<https://doi.org/10.1007/978-3-031-46238-2>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Paper in this product is recyclable.

Contents

Generative AI as a Supportive Tool for Scientific Research	1
Abraham Itzhak Weinberg	
Creating Ad Campaigns Using Generative AI	23
Ahmet Bulut and Bariş Arslan	
Unlocking the Potential of Generative Artificial Intelligence in Drug Discovery	37
Virgilio Romanelli, Carmen Cerchia, and Antonio Lavecchia	
Privacy in Generative Models: Attacks and Defense Mechanisms	65
Maryam Azadmanesh, Behrouz Shahgholi Ghahfarokhi, and Maede Ashouri Talouki	
Generative Adversarial Network for Synthetic Image Generation Method: Review, Analysis, and Perspective	91
Christine Dewi	
Image Rendering with Generative Adversarial Networks	117
Fayçal Abbas, Mehdi Malah, and Ramzi Agaba	
Dsmk-DcSeg-Lap, a Generative Adversarial Network Guided by Dark-Chanel and Segmentation to Smoke Removal in Laparoscopic Images	137
Hugo Moreno, Sebastián Salazar-Colores, Luis M. Valentín, and Gerardo Flores	
Generative AI Use in the Construction Industry	161
Gozde Basak Ozturk and Fatih Soygazi	
Generative AI Applications in the Health and Well-Being Domain: Virtual and Robotic Assistance and the Need for Niche Language Models (NLMs)	189
Graeme Revell	

Generative Adversarial Network Based Deep Learning Method for Machine Vision Inspection	209
Hao Wu	
Generative Adversarial Networks for Stain Normalisation in Histopathology	227
Jack Breen, Kieran Zucker, Katie Allen, Nishant Ravikumar, and Nicolas M. Orsi	
Augmenting Data from Epileptic Brain Seizures Using Deep Generative Networks	249
Jean-Philippe Thivierge	
Can Generative Artificial Intelligence Foster Belongingness, Social Support, and Reduce Loneliness? A Conceptual Analysis	261
Bianca Pani, Joseph Crawford, and Kelly-Ann Allen	
The SEARCH for AI-Informed Wellbeing Education: A Conceptual Framework	277
Kelly-Ann Allen, Margaret L. Kern, Joseph Crawford, Michael Cowling, Duyen Vo, and Lea Waters	
Generative AI to Understand Complex Ecological Interactions	293
Hirn Johannes, Sanz Verónica, and Verdú Miguel	
On the Effect of Loss Function in GAN Based Data Augmentation for Fault Diagnosis of an Industrial Robot	309
Ziqiang Pu, Chuan Li, and José Valente de Oliveira	
Underwater Acoustic Noise Modeling Based on Generative-Adversarial-Network	351
Junfeng Wang, Mingzhang Zhou, Yue Cui, Haixin Sun, and Guangjie Han	
How Generative AI Is Transforming Medical Imaging: A Practical Guide	371
Khaled ELKarazle, Valliappan Raman, Patrick Then, and Caslon Chua	
Generative AI in Medical Imaging and Its Application in Low Dose Computed Tomography (CT) Image Denoising	387
Luella Marcos, Paul Babyn, and Javad Alirezaie	
Generating 3D Reconstructions Using Generative Models	403
Mehdi Malah, Ramzi Agaba, and Fayçal Abbas	
ChatGPT Implementation in the Metaverse: Towards Another Level of Immersiveness in Education	421
Michael Agyemang Adarkwah, Ahmed Tlili, Boulus Shehata, Ronghuai Huang, Prince Yaw Owusu Amoako, and Huanhuan Wang	

Contents	vii
Generating Artistic Portrait Drawings from Images	437
Ran Yi, Yong-Jin Liu, Yu-Kun Lai, and Paul L. Rosin	
AI Deep Learning Generative Models for Drug Discovery	461
Qifeng Bai, Jian Ma, and Tingyang Xu	
3D Generative Network	477
Ran Song, Hao Zhang, and Wei Zhang	
The Economics of Generative AI	491
Stanislav Ivanov	
Plant Data Generation with Generative AI: An Application to Plant Phenotyping	503
Swati Bhugra, Siddharth Srivastava, Vinay Kaushik, Prerana Mukherjee, and Brejesh Lall	
Generative Models for Missing Data	537
Huiming Xie, Fei Xue, and Xiao Wang	
Infrared Image Super-Resolution via GAN	565
Yongsong Huang and Shinichiro Omachi	
Generative AI for Fire Safety	577
M. Hamed Mozaffari, Yuchuan Li, and Yoon Ko	
A Multi-scale Convolutional Autoencoder with Attention Mechanism for Fault Diagnosis of Rotating Machinery	601
Zihao Lei, Hongguang Yun, Feiyu Tian, Guangrui Wen, and Zheng Liu	

Generative AI as a Supportive Tool for Scientific Research



Abraham Itzhak Weinberg

Abstract This chapter Abraham aims to bridge the gap between the theoretical potential of Generative AI (GAI) tools, such as Generative Pretrained Transformer (GPT), and their practical applications as supportive tools for scientific research. The chapter provides approaches and techniques for leveraging GAI to address research challenges and activities. It describes common research tasks and provides guidance on how to use GPT to solve them. To the best of our experience, at the current stage, the integration of researchers and GPT has the potential to yield better results than either could achieve alone. Furthermore, the increasing availability of GPT tools suggests that the synergy between the two will continue to improve research output quality and save time. The key to successful integration lies in the appropriate use of GPT that can be achieved by directing AI tools to solve tasks effectively and using prompt engineering techniques.

1 Generative AI History and Evolution

The exponential expansion of GAI models and their incursion into various domains has been observed in recent years. As the name implies, GAI models create content that even their developers have not considered. Their outputs include textual material, innovative graphics and pictures, chat responses, distinctive object designs, programming, synthetic data, and, sadly, deep fakes [1]. Some GAI models employ human interfaces that replicate human language answers via conversation. This feature closes the gap between the human and computer interfaces. Human interaction, as well as the ability of GAI models to respond quickly to time-consuming tasks, are likely catalysts for the GAI revolution and appeal.

It seems that in addition to their positive potential to contribute to their users, GAI-based tools encapsulate destructive potential. Although GAI models generate their outputs without human intervention, the guidance and required results are controlled by humans. The borderline between a meaningful contribution to destructive output

A. I. Weinberg (✉)
AI-Weinberg, AI Experts, Tel Aviv, Israel
e-mail: aviw2010@gmail.com

is very thin, owing to the powerful potential of GAI models. The GAI output is known as AI Generated Content (AIGC). Generative AI is a class of Artificial Intelligence (AI) models. At the time of writing these lines, GAI models' cores are mainly based on Generative Adversarial Networks (GANs) and Transformer based models such as GPT-3, GPT-3.5 and GPT-4.

GAI applications can be classified as open-source, private, and closed-source GAI [2]. Open source is more relevant to specific enterprise needs and applications, whereas private and closed-source GAI are more general-oriented solutions to AI based on Large Language Models (LLM). Domain-specific sources by their nature are solution-focused, and as a result, their trustworthiness level is higher.

Another way to taxonomize the GAI is by its maturity level of intelligence in solving usecases. According to Schuller et al. [3], classes can be advisory, assistive, cooperative, augmentative, digitally autonomous, or physically autonomous. The advisory class category had the lowest level of intelligence, where the GAI provided domain-specific responses to user prompts. Models categorized as assistive offer automatic task execution to achieve user goals. Models that fall into the cooperative category collaborate with users to achieve the required tasks. Augmentative models enable tasks to be accomplished by bridging the gap between user knowledge and the knowledge required to solve the problem. Digitally autonomous models assist users in accomplishing tasks by making decisions independently in digital environments. The final category, physically autonomous, assists the user in solving their tasks in a physical environment.

The history of GAI is not long. In this chapter, we discuss the scientific research aspect of GAI. The vast majority of research data are based on numerical and textual data. In addition to numerical and text, GAI is used for applications in image and video models, as well as multimodal aspects that combine between the areas. In the context of this chapter, we focus mainly on the textual aspects of GAI. According to Zhang et al.[4] the first meaningful GAI text model was published in June 2017 by Google's researchers known as "Attention is all you need" [5]. This paper is considered a cornerstone of Transformers for AI text-based models. The next most relevant event for textual GAI occurred in 2018. In June 2018, OpenAI released the first GPT model trained with 117 million parameters [6] and in November, Google released Bidirectional Encoder Representations from Transformers (BERT) model. In 2019, a new version of GPT was released (GPT-2) trained with 1.5 billion parameters [7] and Baidu released the Enhanced Representation through Knowledge Integration (ERNIE) model, which is considered to be BERT's rival [8]. After six months, Baidu released a new version of ERNIE - ERNIE 2.0, which outperformed BERT. In October 2019, Google implemented BERT in its search engine and demonstrated a semantic search engine known as Sentence-Bert [9].

In 2020, GPT-3 was released and trained with 175 billion parameters [10]. One impressive presentation was the automatic generation of columns in the Guardian British daily newspaper [11]. In June 2021, Microsoft released Copilot with its subsidiary GitHub. Copilot is a code-completion tool based on 54 million GitHub repositories [12]. At the end of the same year, Baidu released a third version of ERNIE. ERNIE 3.0 was trained on structured knowledge graph data to help the model achieve

coherent responses [13]. Approximately half a year later, in April 2022, Google released Pathways Language Model (PaLM) trained on 540 billion parameters that outperformed GPT-3 on some NLP tasks [14]. In the same month, Meta released its LLM solution for an Open Pre-trained Transformer (OPT) with 175 billion parameters [15]. The BigScience Large Open-science Open-access multilingual language Model (BLOOM) was released in July 2022 [16]. It is an autoregressive LLM developed and trained with 176 billion parameters. BLOOM uses vast amounts of data and text with industrial-scale computational resources. It was developed by the French National Center for Scientific Research (CNRS). In addition, it was an open source platform that ran on the HuggingFace platform. In November 2022, OpenAI released ChatGPT [17, 18] based on a new version, GPT-3.5, which provided more accurate responses and eliminated toxic output than GPT-3 [19]. ChatGPT allows users to interact with chatbots for human-like conversations.

In February 2023, Google released BARD. BARD is Google’s answer to the ChatGPT. BARD meaning is poet. It is based on the Language Model for Dialogue Applications (LaMDA) Google’s core model. It outperformed ChatGPT in answering questions with specific information [20]. In March 2023, OpenAI released ChatGPT-4.0, with one trillion trained parameters (six times larger than that of GPT-3). ChatGPT 4.0 is multimodal. It can use different types of inputs, such as text and images, and synthesize them between them [21]. In April 2023, RedPajama released a 1.2 Trillion token dataset following the LLaMA recipe. This history is a glimpse into the fascinating, fast-growing world of GAI. It shows enormous development efforts and potential applications in general, and in the scientific field in particular. In the following section, we will take a closer look at how GAI models work and examine some of the key challenges and opportunities associated with this rapidly evolving field.

2 How Does GAI Models Work?—Training GAI Models: Data Sources and Techniques

In this section, we describe the main phases of GAI training and their sources. This section provides a high-level overview of GAI chat models and their main training phases. GAI chat models create a vast spectrum of outputs, such as video, audio, image, code, graph, table, and text. The generation process comprises two main steps: extracting information from human instructions and generating the required content [22]. There are three main phases of GAI training: pre-training, Supervised Learning (SL), and Reinforcement Learning from Human Feedback (RLHF) [17, 23]. The last category is dedicated to the interaction between humans and the model. GAI-trained sources are diverse, and their amount and diversity show an increasing trend. Just to mention a few, the trained sources are based on Internet sources such as Wikipedia and web crawler data as well as news, books, academic papers, music, and code. GAI chat instruction-based models, such as GPT, predict the next step or result

by learning from pre-trained data. Typically, GPT models use an interactive dialogue known as a prompt. This prompt uses a sequence of requests, responses, or dialogue between questions and answers. This dialogue can be modelled as a sequence of objects such as words or sentences. The core mechanism is based on Transformers architecture. The core of the Transformer uses a self-attention mechanism to relate different parts of its input sequence [5]. Typically, Transformers include Encoders and Decoders to generate hidden representations and output sequences, respectively, [22]. Transformer-based models can be divided into several classes: encoder-based (such as BERT), decoder (such as GPT), and encoder-decoder (such as T5 and BART).

The pretraining phase can be classified into two types of training tasks: AutoRegressive (AR) language modeling and masked language modeling. AR language modeling predicts the probability of the next token, given the previous tokens. The masked modeling approach target predicts the probability of a masked token given context information. The following equation is fundamental to the GAI LLM AR-based sequence [5]:

$$P_{\Theta}(X_{t+1} = x_{t+1} | x_1, \dots, x_t) \quad (1)$$

The above equation predicts the next step (or data) that has the highest probability, given the previous steps (x_1, \dots, x_t) . Usually, in LLM x_i refers to tokens such as characters, subwords, or words. Θ is a vector of parameters that maximizes this probability. In recent GAI models, optimization includes trillions of parameters. There are several approaches to producing a generative output, such as GANs, Variational Auto Encoders (VAE), flow AR networks, and diffusion models [24, 25]. Each has its own benefits and drawbacks.

The GANs are based on a zero-sum game between two machines: the generator and discriminator for producing the optimized output. GANs are based on the adversarial learning of the generator. The VAE is based on an autoencoder architecture that includes encoder and decoder layers. Between them, a latent space layer provides probabilistic descriptions of the observations. The VAE relies on a surrogate loss, and its target function maximizes the variational or Evidence Lower BOund (ELBO) [26]. Flow-based models learn the distribution by sampling data explicitly and using log likelihood as an activation function for estimating the parameter [27]. AR flow based models work in noramilize the dimensions into a vector variable for standing under the condition of preceding dimensions [28]. AR uses flow and inverse layers to transform a distribution [29]. Diffusion models are based on the idea of a random walk process [30]. They used noise to estimate the probability density function. The diffusion models add Gaussian noise gradually at each layer and reverse it at the end.

Following the aforementioned training steps, there were two additional training phases. Their main purpose is to help the GAI instructive chat cope with the alignment problem [31]. The alignment problem is that the pre-training phase is not necessarily aligned with human language richness and meaning. The next training phases are aimed at closing the human-model gap by increasing its usefulness and trustworthiness. The SL phase is based on datasets of human responses to questions and

tasks. Each pair of prompt and generated answers was rewarded in a human manner. In practice, a human-manner reward vector $\{R_1, \dots, R_n\}$ is created for each prompt $\{P\}$, and the GAI model's generated answers $\{a_1, \dots, a_n\}$ are created in the previous pretraining phase. This is Imitation Learning or Behavior Cloning that mimics the actions of experts conditioned on the prompt $\{P\}$.

The next training step is RLHF. Its main purpose is to solve the distributional shift. The shift stems from the difference between the distribution of the training and inference data. To solve this issue, the model was trained to act rather than observe an expert [32]. This was implemented using Reinforcement Learning (RL) Policy Optimization (PPO) [33]. The reward function was based on human preferences [34]. The RL policy model actions are based on two inputs: the pretrained GAI model state and SL-based reward model value. A penalty was added to the PPO model to avoid overfitting between the SL reward model and real human results [35]. The penalty is based on the Kullback-Leibler divergence $R_{TOT} = R_{SL} - \lambda_{KL} D_{KL}(\Pi_\theta | \Pi_{\theta_0})$. Where R_{SL} is the SL reward and λ is the RL PPO smoothing parameter [33]. D_{KL} is the Kullback-Leibler divergence penalty for Π policy with θ pre-trained parameters. This penalty was considered for each RL training episode [22].

The described process stages enable the Chat GPT model to achieve a high level of accuracy, side-by-side to human-oriented responses to user requests. After gaining an understanding of the mechanism behind GAI, we proceed to compare it with other approaches in the next section.

3 The Difference Between Conventional ML and Deep Learning (DL) Systems, Generative AI Models and Prompt Based ML (PML)

In previous section, we mentioned the evolution of GAI and their history. GAI is an approach that evolved only in recent years. In this section, we will sharpen the difference between the GAI and their usages in Prompt based ML (PML) to Traditional Machine and Deep Learning (TML). PML is a machine learning (ML) strategy that uses pretrained LLM models and enables interaction with the user usually by prompt [36]. Usually, PML use GAI in order to solve it tasks.

The LLM can be classified into two main classes: The base LM that predicts the next object (such as a word) and Instruction Tuned LLM that follows and replies to user's instructions and requests [37] that also relates to PML. TML models are considered as building blocks for GAI with a longer profound history in comparison to PML and GAI. The difference between TML and GAI resides in the process and the destined tasks. The main difference between the two relies on their focus. The goal and focus of TML is to understand the relations between the data and as a result to be able to predict the target values. GAI on the other hand, focus on creating and synthesising new data that is similar to the source data and stands under constraints.

Usually TML consists of three main stages: labeling the source data, developing a model: training and testing, and finally deploying the model in production. PML on the other hand, consists on two main stages: the prompt model and deployment. It usually shortens the processing time by magnitude in comparison to TML. To the best of our knowledge, the first PML phase prolongs between minutes and hours, and the deployment phase lasts between a few hours to days. On the other hand, TML labeling phase can take weeks or even months, the same magnitude of time for the model development, and the deployment phase can take between day to months. The known theorem in ML “No Free Lunch” (NFL) [38] is applicable here. The reason for this relies on the fact that PML are already pretrained. Practically the PML models are pretrained with trillions of parameters thousands of hours of training on large amounts of different data sources. This is one of the reasons that enables GAI models to support few shots or zero shot learning techniques.

Additional point that distinguishes between PML and TML is its iterative learning capability. This enables the PML user to update his request and get a relevant answers. On the other hand, iterative learning is not always supported by TML models and sometimes the model has to be trained from scratch. The differences between the TML and PML can be distinguished in the their results as well. As mentioned before, TML usually has a predefined task to solve, whereas the PML has the flexibility to produce output that answers the user needs. An example of this flexibility can be achieved by asking the PML to use a different role to answer the same question. Wearing different hats to answer user requests is supported by the GAI capabilities.

The differences between TML, PML and GAI can help understanding the quantum jump enabled by GAI models.

4 GAI Model Usages and Tools

Using GAI models can enhance the scientific work of researchers. Their work is characterized by the use of a large amount of material from a wide variety of sources. Furthermore, the sources they use may originate from fields other than their expertise. The final scientific product is typically paper. This study must go through a rigorous review process. Recently, GAI-based tailored tools that can assist researchers along with research phases have become increasingly evident. AI-based tools for scientific use are trained using scientifically oriented data. The models interact with the user by a prompt-based interface combined with add-ins, which supports a variety of researcher tasks. We can mention tools such as SciSpace [39], Humata [40], Explainpaper [41], Askyourpdf [42] and Gamma [43]. Using these tools, researchers can upload papers and address specific tasks related to the content and specific locations within the text. Other tools enable the creation of presentations and workflows concerning the relevant required materials. In addition, there are GAI tools that enable the transfer between different sources of data, such as from speech to text, text to tables, and graphs.

Given that the majority of research material is text based, our attention in this section is predominantly based on models. The vast majority of research material is based on LLM GAI model Application Programming Interface (API). A general taxonomy of these tools can be based on main tasks such as inference, summarizing, data analysis, text transformation, and expansion [37, 44–47]. Inferring tasks include topic extraction and emotion detection, known as sentiment analysis. Summarizing is one of the researchers popular tasks, it shortens the document reading time by selecting the most important and influential sentences from the whole text. A summary can range from several sentences to a paragraph. Text transformation includes tasks such as spelling correction and translation of text to other languages or sources, such as speech. Text expansion includes tasks such as word and sentence autocompletion and automatic writing of texts. One popular technique for increasing the trustworthiness of the summarization process is to ask the model to use “extraction.” The extraction directs the model to derive a summary from the source sentences rather than to generate them from scratch. In addition, summarization and extraction can be seen visually by creating an automatic presentation based on text, for instance [43].

Additionally, GAI models can use different roles to answer the same question. In the context of scientific research, the user can define, for instance, the role of a reviewer or author and ask the model for relevant questions. Wearing different hats enables the author to look at his work in three hundred sixty degrees and hopefully receive unbiased feedback on the research and the paper. This can shed light on the points and areas that he has not been aware of. In addition, different roles can control format and writing style. Controlling the randomness and generative level of the model can be used to generate several outputs for the same role, such as simulating different reviewers.

In the next paragraphs, we describe typical research workflow stages and demonstrate how the researcher uses GAI for each stage to improve his work.

! Attention

We emphasize that the ideas and approaches mentioned below should be used carefully. Their usage as part of any published material must comply with the published venue instructions. In addition, the researchers should mention that the GAI model was involved in the research process and elaborated on its contribution.

The first stage of the research is to identify and develop the topic. By asking the chat GAI model relevant questions, such as, is the topic updated? How many papers have been published on the research topic? In addition, by inserting the paper headline into the model, researchers can ask questions about the paper’s novelty and potential contribution to the research field. GAI can help find relevant papers and background information that will help the researcher make the right decisions about the research content. Researchers can even ask the assistance of GAI to fine-tune and formulate relevant questions to find the most suitable research topic. Another task

that the GAI model can help is finding the right connection or sequence between papers. The models can evaluate papers using different parameters, as well as their contribution and relevance to the research topic.

Eventually, when the researcher decides to start writing a paper, GAI can help with the process. We divided the process into time workflow stages: before writing, writing, and writing the paper.

4.1 *Pre-writing Stage*

In the pre-writing stage, the GAI can help in finding reference papers and materials. In addition, they can summarize them. GAI chat models can use their attention mechanism to answer research questions based on their pre-trained data as well as the researcher's uploaded materials. The user can ask the model to find relevant answers across all references or focus on a specific one. GAI can help explain texts, graphs, and images. Some researchers have asked the GAI to produce code on a given paper. Alternatively, they used natural language instructions to produce codes for their research. In the market, we can find GAI code generation tools, and their number is in a growth trend. These tools make coding much easier by harnessing the powerful inherent capabilities of GAI for creating lines of code [48]. In addition, text-to-speech applications can read the required texts or paragraphs while the researcher dictates instructions.

4.2 *Writing Stage*

The writing phase is considered to be one of the most time-consuming parts of research. Publication of the research is the front window and provides a glimpse of the researcher's work while demonstrating his contribution to science. The GAI can accompany the researcher at this stage and optimize it. GAI models can help in writing, editing, and styling of written materials. These models can also help in tasks such as the autocomplete of words, finding the most appropriate words, and rephrasing. These tasks can be helpful for many researchers who publish in a language that is not their native language. The GAI model can help in styling tasks, such as changing the format of tables and graphs, and automatically creating graphs based on tables and data. There are GAI-based models and tools that help to divide the text into consistent paragraphs. GAI tools such as Speechmatics [49] help researchers transfer their voice to text with a high level of accuracy even in noisy environments. In this way, the researcher can save writing time or parallelize his work. In addition, GAI tools can help create graphs in a wide variety of styles using different types of data sources. Converting equations, formulas, tables, images, and text, and integrating them into other formats such as Latex format [50] can be time-consuming tasks that GAI models can assist with.

4.3 Post-writing Stage

After completing writing the paper the researcher can use the GAI model for getting a feedback. Researcher can use Role Prompting technique. Role prompting enables a GAI to take a specific role while responding to its user. The researcher can ask the GAI model to take the role of reviewer, for instance, and find the weak points in the paper and/or in the research. In addition, the researcher can use the Attention technique to ask relevant questions. The Attention technique enabled by the GAI model provides exact answer locations for the questions required in the paper body. The researcher can test the readability level of the papers. Tasks such as summarizing the paper can be helpful in ensuring that the main points of the research are mentioned in the text. An additional feedback method is keyword extraction. Keyword extraction can be based on Zero Shot technique. GAI models can provide answers to questions without prior knowledge. Keyword extraction can indicate whether the paper includes the main research subtopics and if the amount of information for each subtopic is well distributed. This process can also be helpful for the user to decide on the right venue for the paper. In addition, GAI can help ensure a consistent writing style. This need intensifies for several contributors.

Chat GAI models can help check and detect plagiarism. The GAI model can use the vast amount of data it was trained on and compared between the sources and the tested material. For such a comparison, GAI can use Natural Language Processing (NLP) functionality, such as synonyms and rephrasing, to compare content that looks differently only from a semantic perspective, although their content is the same.

An additional related and tricky functionality uses the GAI model as an AI writing detector [51]. At the moment of writing on these lines, the question is considered a fundamental issue, especially in academia. From a philosophical perspective, this question resembles an oxymoron. Can the GAI detect whether it or other machines create a specific text or a human? Although this challenge is considered an important issue and has a high impact in many fields, GAI still faces several gaps in its solution [52].

In this section, we discuss several techniques and approaches that researchers can use wisely during the research stages. In the next section, we elaborate on the techniques for optimizing GAI model performance.

5 Optimizing and Utilizing GAI Models Through Prompt Engineering

In this section, we briefly describe the Prompt Engineering concept and several methods for fine-tuning models. Fine-tuning is a method for tailoring PML GAI-based models to achieve optimal performance and results for a specific task [53, 54]. However, not all models enable or expose this functionality. When enabled, a fine-tuning process can improve the model performance for a destined task. Fine-tuning has an

impact on common GPT model tasks such as classification and conditional generation, where the model generates data according to predefined conditions. The impact of fine-tuning is demonstrated in popular classification usecases such as enforcing the model to make true statements. Fine-tuning is an additional approach for resolving the alignment challenge of closing the gap between GAI models and human language richness and meaningfulness. In the conditional generation category, we can find usecases such as the creation of support chatbots, entity extraction in sentences, and question answering.

As these models are trained for general-purpose tasks, they are built on trillions of parameters and a diverse range of tasks. Fine-tuning harnesses the flexibility of a model in a specific application. A good fine-tuning process can lead to higher quality results and metrics, decrease the number of training prompts, and lower latency. In addition, fine-tuning helps the model converge faster and, as a result, uses fewer tokens to train the prompt. Moreover, in few-shot tasks, the user can provide fewer examples per prompt if any.

Recently, several techniques have been proposed to improve the performance of GPT-4 models in zero-shot and few-shot tasks. One such technique, called instruction tuning, was introduced by Peng et al. [55] and involves the generation of fine-tuning instructions to optimize the model performance. Another approach, proposed by Liu et al. [53], is called P-tuning, and is based on trainable prompt embeddings. According to their results, P-tuning improves model performance in both few-shot and supervised settings while eliminating the need for prompt engineering.

To fine-tune a language model, the process is typically divided into stages, such as preparing the training data, fine-tuning the model, and testing the trained fine-tuned model. In this section, we discuss the following fine-tuning parameters: temperature, max tokens, top p, frequency, and presence penalty. Temperature controls the randomness of the model. The higher the value of, the more creative the model. Creativity comes at the expense of reliability and predictability. Usually, the temperature range is between zero and one, where one represents the highest level of creativity. The model outputs are based on the final neural layers: a logit function that includes the unnormalized values, and the softmax function that converts its values into probabilities. Softmax determines whether the logit value passes with or without modification, based on its probability values. As a result, a temperature close to zero indicates that there is a higher probability that the newly generated token will resemble the previous ones, and the results are more deterministic.

The max tokens parameter determines the GAI model maximal number of generated tokens in the GAI model. Tokens typically consist of words or characters. Increasing the max tokens value provides a model with greater flexibility for generating new output.

Top P also called Nucleus Sampling is a technique that samples values from model output at the inference time by determining a threshold [47]. Practically, it chooses values where the Cumulative Distribution Function (CDF) has a probability higher than the determined threshold. Top P controls the GAI model randomness. The lower the value of, the more candidates that can be generated.

The Frequency and Presence penalty parameters use penalties to control the repetitiveness of GAI model predictions. The first penalizes the model in cases in which the same words have already been generated, whereas the Presence penalty encourages the model to use new words. The Presence penalty penalizes the model by lowering the probability that words have already been generated. This approach is complementary to the Frequency penalty because it does not depend on the previous frequency of the words in the generated text.

In this section, we have discussed several parameters that can be used to customize GAI models according to specific needs. These parameters offer scientists greater control over the behavior and performance of the models. In the next section, we will explore how users can evaluate the performance of GAI models.

6 Generative Models Evaluation and Metrics

In the preceding sections, we discussed the primary scientific research tasks and how GAI models can aid in their accomplishments. The output of a GAI model is not a precise mathematical solution, and the assessment of its quality depends on user decisions and feedback. In this section, we present the main metrics and approaches for evaluating the performance of GAI models.

ML evaluation metrics are widely used and are considered essential components for comparing models and improving model performance. The ML evaluation process is relatively straightforward because there is a ground truth for comparing the outputs of the model. Some of ML commonly used evaluation metrics are training time, accuracy, confusion matrix, logarithmic loss, Area Under Curve (AUC), F1 score, Error based metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE), and visual graphs such as the Receiver Operating Characteristic curve (RUC) [56].

GAI model evaluation is more challenging because its results are qualitative and there are several valid answers to the same task. Moreover, the model usually does not provide all the options. The evaluation of the GAI results usually depends on the eye of the beholder.

GAI model evaluation can be based on soft metrics as well as quantitative metrics. Soft metrics consider the factors that depend on human perception. This is tricky because different people can perceive the results differently. Some human-oriented metrics are quality, creativity, and human language compliance [57]. LLM GAI metrics can use language metrics in addition to general GAI evaluation metrics. For example, the richness of vocabulary and diversity. There are also metrics that indicate the reliability, trustworthiness, and fidelity of results. Another metric that can be employed is the comprehensiveness of the answers in covering the content and topics of a given task.

The first metric that can be used for GAI models is the execution time. This can be computed as the time required for the GAI model to respond to the user's request.

An additional approach for this metric is the elapsed time for receiving an acceptable answer by the user or, alternatively, a correct answer.

The quantitative metrics were based on formulas and measurements. Some commonly used quantitative GAI evaluation metrics [58–61] have been defined for image-generation models: Inception Score (IS), Frechet Inception Distance (FID), Contrastive Language Image Pretraining (CLIP) score [62, 63] for evaluating Text Image similarity, and L2 or L1 norms for assessing content preservation.

Popular metrics for language generation models are Recall and Precision scores. Recall and Precision scores are metrics commonly used for ML and retrieval models. Precision measures the accuracy of positive predictions by dividing the true positive cases by the sum of true positive and false positive. Recall uses the same numerator and divides it by the sum of the true positive and false negative. Hence, Precision measures the accuracy of positive predictions and is considered a measure of the quantity or type I error. Recall, on the other hand, measures the completeness of positive predictions and is considered a complementary measure of quantity or type II error. In addition, there are relevant metrics for LLM GAI models such as Recall-Oriented Understudy for Gisting Evaluation (ROUGE) for summarization evaluation [64] and BiLingual Evaluation Understudy (BLEU) scores for translation closeness between languages [65], BERTScore [66], METEOR [67] and Uniform Manifold Approximation and Projection (UMAP), which can be used as visualizing tools [68]. BERTScore calculates the relation between the reference and generated text using contextual embeddings of words rather than exact matches used in BLEU and ROUGE. METEOR calculates the quality of the generated text using the relations between precision and recall. Model reference is an essential input for computing ROUGE and METEOR. Another approach involves using UMAP [68] as a visualization tool to compare the generated text by creating clusters that represent similar semantic meanings.

Zheng et al. [69] distinguish between LLM models to ChatGPT evaluations. LLM evaluation includes benchmark metrics such as Beyond the Imitation Game (BIG-bench) [70], Massive Multitask Language Understanding (MMLU) [71], and TruthfulQA [72] for question answering. Other approaches, such as the Holistic Evaluation of Language Models (HELM) [73] provide metrics that indicate the transparency level of the LLM model.

Mahowald et al. [74] differentiate between formal competence knowledge of linguistic rules and functional competence understanding and using language in the world. There are several more LLM metric approaches that are relevant to specific tasks, such as autonomous research [75] with a focus on the fields of biology, chemistry, and computer programming. We can find evaluation methods for text to sql tasks [76]. Armengol et al. [77] propose additional metrics for evaluating LLM multilingual capabilities.

Performance-based metrics can be tailored to specific types of tasks, such as solving mathematical problems [78]. According to Zheng et al. [69] we can categorize ChatGPT evaluation into the following: general evaluation methods and metrics [79], understanding abilities [80], bug and error fixing performance [81], Out Of Distribution (OOD) zero-shot-based behaviors [82], translation [82], and Question

Answering performance [57]. In addition, we can find approaches to evaluate the quality of a model, such as reasoning [83], factual accuracy [79], and ethical implications [84].

As previously mentioned, the evaluation of GAI models is an ongoing process that requires the use of both qualitative and quantitative metrics to assess performance across different tasks. While each approach can be useful, combining them can yield better evaluation results. In the next section, we will discuss an important issue related to model evaluation: model trustworthiness.

7 Does Chat GAI Models Cheat? - GAI Pitfalls and Their Proposed Solutions

The GAI model, by its design, has generative characteristics and has the potential to produce unpredictable results. This is one of the reasons that their designers notify users of the cautious usage of their results. This characteristic does not imply a constant trend of AI trusrtworthiness. Trusrtworthiness is acute in the field of scientific research. Zheng et al. [69] categorize the GAI model's truthfulness challenges into the following classes: comprehension, factualness, specificity, and inference. Comprehension refers to the model's capability to understand a user's task context and intention. Factualness is a model's ability to support its answers based on facts. Specificity is the extent to which a model can provide the correct level of specificity. Usually, the required specificity relies on an optimized level between the general and specific answers. Inference is the model's capability to deduce facts effectively given that it has the necessary knowledge to answer the required question.

In this section, we provide several ideas for increasing GAI model truthfulness. In practice, there is no way to trust the factual knowledge produced by the generative model [85, 86]. This intensifies in the question-answering task, which is at the heart of chatting models. Jiang et al. [87] proposed and developed approach for increasing calibration in order to improve the quality of answers of GAI question answering models.

In the previous section, we discussed the GAI model parameters called the Model Temperature (MT) or simply Temperature. This parameter controls the predictability of the model. There is a negative correlation between the model's reliability and creativity. The closer MT is to zero, the more reliable the model is.

The next approach that can be used to increase the GAI output trustworthiness is called a Closed Book (CB) [23]. CB question-answering is an approach that uses answers to questions without an external knowledge source. Alternatively, the CB approach forces the GAI model to base its output on a specific or reliable source of knowledge. The open-domain or open-book approach aims to answer questions based on large-scale data and documents. The name "Closed Book" originates metaphorically from the student closed book exam approach. In the CB test, the student must memorize the relevant answers based on the information learned before taking the

test [88]. Examples of CB models include GPT, T5, and BART [85, 89]. CB models are pretrained on a large-scale corpus and use or imply their learned parameters on a specific task [90].

An approach that can be used alone or in combination with CB is called Contrastive Learning (CL) [91]. The models use the CL technique to evaluate the connection between pairs of questions and optional answers. The CL characteristics of self-supervised, task-independent techniques allow the learning of positive and negative connections between questions and answers, even without labels or data, as in the CB case. In this manner, the CL finds the best answer from all candidates to the required question. The CL is ideal for zero-shot tasks. A zero-shot is a machine learning task where, as in the CB case, the machine identifies classes and replies to tasks that have not been seen before. Using CL, the GAI can differentiate between reliable and unreliable answers.

An additional approach recently proposed for improving model trustfulness is Iterative Truth Injection (ITI) [92]. ITI alters the model’s activation during inference by applying a set of instructions across a limited number of attention heads with high probing accuracy for truthfulness. This method saves training time, annotation resources, and computation compared to RL-based techniques such as RLHF.

The next, proposed approach for increasing GAI trustworthiness relies on an ensemble of models or multiple interacting agents [93]. The Ensemble approach uses several machine learning models and extracts a decision based on the integration between them. Usually, in the machine learning field, an Ensemble yields better predictive performance than each of the Ensemble’s individual models.

The Ensemble proposed approach is also used in GPT next-generation models such as [94, 95] Chain-of-Thought (CoT) approach based on Self-consistency and Diverse Verifier on Reasoning Step (DiVeRSe) discussed in the next section in more detail. The proposed Ensemble approach consists of running the same task on several GAI models and comparing the results. If there is consent between them, there is a higher probability that the answer will be trustworthy. This comparison can be based on semantic similarity (also known as paraphrase detection) [6].

The Ensemble approach can use several strategies to make decisions. Usually, an Ensemble uses voting strategies such as majority voting. The results can be based on GAI models trained exclusively on data sources. To imply that the GAI models answer the majority voting technique, there is a need to obtain consent or similarity between their outputs. This can be achieved by asking each GAI model if the output is similar to the other outputs. An alternative approach can use the GAI model, which will be considered as a judge and will not take part in the voting process.

An approach to increase trustworthiness relies on explainability. Explainable AI (XAI) is a framework that aims to bridge the gap between humans and machines. This is done by interpreting the AI model’s output in a human-understandable manner. One of the straightforward outcomes of XAI is trustworthiness. There are several ways to explain the results in the GAI context. The first is the most fundamental reference. Asking the GAI model to find the sources of answers helps the user decide how much to rely on. The provenance of traceability can help the user add relevant citations and

references as well as credit to the original creators of the information. The user can also try to find the source of the results using other GAI tools or search engines.

An additional XAI approach relies on interpretability. Interpretability tools use techniques to explain the AI model results to the user. There are XAI approaches that explain the model itself, known as Global Interpretability, whereas others explain the specific output (Local Interpretability) [96]. Interpretability tools can be used as model agnostic for the GAI outputs. XAI can also be used in cases in which the GAI model cannot provide an acceptable explanation of its results. Examples of XAI solutions for specific GAI tasks can be found, such as code generation [97] and attention map GAI visualization tasks. An additional approach that supports GAI XAI is to use a surrogate model or a model simplification technique [96]. XAI surrogate model can explain GAI models that are based on Deep Neural Networks (DNN) and fine-tuned by trillion parameters and aims to find a simpler model that produces similar results. Usually, the surrogate model is clearer to the user and helps him understand how the model arrives at the answer.

This section outlines several approaches that have been proposed to address one of the critical challenges of GAI models: their trustworthiness. In the next section, we will discuss some of the future challenges that GAI is likely to face.

8 Generative AI the Next Generation: Challenges and Opportunities

The GAI market is growing rapidly, with a size of over ten billion USD in 2022, and is expected to grow exponentially over the years [98]. As mentioned before, the growth will be in both the horizontal and vertical directions. GAI is poised to penetrate new market segments, and technological advancements allow its utilization in a wide range of applications. Owing to its rapidly growing learning rate and performance, the GAI can play a significant role in human education and research [99].

Furthermore, GAI has the potential to solve many more tasks efficiently. One of its primary goals is not only to save time and increase efficiency, but also to produce valuable outputs. As previously mentioned, one of the crucial factors that can affect a model is its level of truthfulness and reliability. When this is achieved, there is a greater likelihood that investments in the field will bear fruit and have a stable foundation to build upon.

New GAI models and technologies with the potential to address novel challenges are continuously emerging. In this section, we will discuss two approaches that are considered as next-generation GAI: AutoGPT [100] and CoT prompting or Auto-CoT [94, 95, 101, 102].

AutoGPT [100] is a GPT-based technique that enables the division of a complex task into sub-tasks and runs them step by step. AutoGPT can be combined with few-shot prompting to achieve improved results. The whole process can be done automatically, and hence its name is Automatic (Auto) GPT. CoT [103] technique was proposed by

Google in 2022. As part of the AutoGPT family, it generates a series of intermediate steps that lead to a complete task solution. An instance of the AutoGPT approach in the realm of scientific research could be a task such as experimental design, in which the AutoGPT model can be divided into steps and provides solutions for each step.

AutoGPT is based on two main paradigms [100]: dividing the task into a sequence of step-by-step subtasks and using the trained data to solve them. For example, CoT uses a question-and-answer reasoning chain that leads to an answer. The reasoning chain contributes to its name, the Chain Of Thoughts. To optimize CoT performance, several promoting techniques exist, such as [104, 105]: Zero-Shot CoT, Least-to-most Prompting, Self-consistency and DiVeRSE [94].

Zero-Shot CoT constructs a reasoning path with no need for data or prompt modification and finds the correct solution. This enables step-by-step answers across a wide range of reasoning tasks. Least-to-most Prompting decomposes the task problem into smaller answerable subtasks so that each sub-task result is the input for the next one. Self-consistency is a greedy decoding technique. It is based on majority voting for choosing a route to a solution. DiVeRSE [94] uses a group of prompts to solve the same problem, filters incorrect answers by voting, and verifies each reasoning step.

The AutoGPT solution also offers innovative added value by being able to correct its own solutions and rectify errors [106]. For instance, if AutoGPT does not find the required data file, it uses the model to find it in different directories and updates the relevant sources and code. This section covered the next generation of GPT models, whose beginnings have already emerged on the horizon.

9 Conclusions

This chapter outlines the challenges and techniques involved in implementing GAI in scientific research. We discuss practical research tasks and how GAI can be utilized as a supportive tool to solve them. Additionally, this chapter explains the GAI mechanism and approaches to maximize its performance. Despite the tremendous potential of GAI to aid researchers in scientific activities, there remain significant challenges and gaps that must be addressed. We anticipate that research and development efforts in GAI will lead to improved performance, reliability, and trustworthiness. We believe that in the coming years, GAI will become a fundamental supportive tool that propels science and research to new heights.

References

1. Gozalo-Brizuela, R., & Garrido-Merchán, E. C. (2023). A survey of generative ai applications. arXiv preprint [arXiv:2306.02781](https://arxiv.org/abs/2306.02781).
2. Gade, K. (2023). Two worlds of generative ai: Closed source agi vs. open source narrow ai,” <https://www.linkedin.com/pulse/two-worlds-generative-ai-closed-source-agi-vs-open-narrow-gade/> [Online; accessed 12 May 2023].

3. Schuller, S. (2023). A 6-category taxonomy for generative ai use cases. <https://nuvalence.io/insights/a-6-category-taxonomy-for-generative-ai-use-cases/> [Online; accessed 12 May 2023].
4. Zhang, C., Zhang, C., Zheng, S., Qiao, Y., Li, C., Zhang, M., Dam, S. K., Thwal, C. M., Tun, Y. L., & Huy, L. L. et al. (2023). A complete survey on generative ai (aigc): Is chatgpt from gpt-4 to gpt-5 all you need? arXiv preprint <arXiv:2303.11717>.
5. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems 30*.
6. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training. *OpenAI blog*.
7. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 9.
8. Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X., Zhang, H., Tian, X., Zhu, D., Tian, H., & Wu, H. (2019). Ernie: Enhanced representation through knowledge integration. <arXiv:1904.09223>.
9. Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. <arXiv:1908.10084>.
10. Floridi, L., & Chiriatti, M. (2020). Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30, 681–694.
11. Ugli, M. I. B. (2020). Will human beings be superseded by generative pre-trained transformer 3 (gpt-3) in programming? *International Journal on Orange Technologies*, 2(10), 141–143.
12. Pearce, H., Ahmad, B., Tan, B., Dolan-Gavitt, B., & Karri, R. (2022). Asleep at the keyboard? Assessing the security of github copilot's code contributions. In *IEEE symposium on security and privacy (SP)* (pp. 754–768). IEEE.
13. Sun, Y., Wang, S., Feng, S., Ding, S., Pang, C., Shang, J., Liu, J., Chen, X., Zhao, Y., Lu, Y. et al. (2021). Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. <arXiv:2107.02137>.
14. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. (2022). Palm: Scaling language modeling with pathways. <arXiv:2204.02311>.
15. Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. (2022). Opt: Open pre-trained transformer language models. <arXiv:2205.01068>.
16. Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., Gallé, M., et al. (2022). Bloom: A 176b-parameter open-access multilingual language model. <arXiv:2211.05100>.
17. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems 35*, 27 730–27 744.
18. Roose, K. (2022). The brilliance and weirdness of chatgpt. The New York Times.
19. Hagendorff, T., Fabi, S., & Kosinski, M. (2022). Machine intuition: Uncovering human-like intuitive decision-making in gpt-3.5. <arXiv:2212.05206>.
20. Rahaman, M., Ahsan, M., Anjum, N., Rahman, M., Rahman, M. N., et al. (2023). *The ai race is on! google's bard and openai's chatgpt head to head: An opinion article*. Md Nafizur, The AI Race is on: Mizanur and Rahman.
21. Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. (2023). Sparks of artificial general intelligence: Early experiments with gpt-4. <arXiv:2303.12712>.
22. Cao, Y., Li, S., Liu, Y., Yan, Z., Dai, Y., Yu, P. S., & Sun, L. (2023). A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt. <arXiv:2303.04226>.
23. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.

24. Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Shao, Y., Zhang, W., Cui, B., & Yang, M.-H. (2022). Diffusion models: A comprehensive survey of methods and applications. [arXiv:2209.00796](https://arxiv.org/abs/2209.00796).
25. Weng, L. (Jul 2021). What are diffusion models?" *lilianweng.github.io*. [Online]. Available <https://lilianweng.github.io/posts/2021-07-11-diffusion-models/>.
26. Odaibo, S. (2019). Tutorial: Deriving the standard variational autoencoder (vae) loss function. [arXiv:1907.08956](https://arxiv.org/abs/1907.08956).
27. Ho, J., Chen, X., Srinivas, A., Duan, Y., & Abbeel, P. (2019). Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International conference on machine learning* (pp. 2722–2730). PMLR.
28. Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., & Welling, M. (2016). Improved variational inference with inverse autoregressive flow. *Advances in Neural Information Processing Systems 29*.
29. Huang, C.-W., Krueger, D., Lacoste, A., & Courville, A. (2018). Neural autoregressive flows. In *International conference on machine learning* (pp. 2078–2087). PMLR.
30. Cao, H., Tan, C., Gao, Z., Chen, G., Heng, P.-A., & Li, S. Z. (2022). A survey on generative diffusion model. [arXiv:2209.02646](https://arxiv.org/abs/2209.02646).
31. Glaese, A., McAleese, N., Trebacz, M., Aslanides, J., Firoiu, V., Ewalds, T., Rauh, M., Weidinger, L., Chadwick, M., Thacker, P., et al. (2022). Improving alignment of dialogue agents via targeted human judgements. [arXiv:2209.14375](https://arxiv.org/abs/2209.14375).
32. Ross, S., & Bagnell, D. (2010). Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 661–668). JMLR Workshop and Conference Proceedings.
33. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. [arXiv:1707.06347](https://arxiv.org/abs/1707.06347).
34. Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems 30*.
35. Gao, L., Schulman, J., & Hilton, J. (2022). Scaling laws for reward model overoptimization. [arXiv:2210.10760](https://arxiv.org/abs/2210.10760).
36. Jiang, E., Olson, K., Toh, E., Molina, A., Donsbach, A., Terry, M., & Cai, C. J. (2022). Promptmaker: Prompt-based prototyping with large language models. In *CHI conference on human factors in computing systems extended abstracts* (pp. 1–8).
37. Andrew, N., & Isabella, F. (2023). Chatgpt prompt engineering for developers. <https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/> [Online; accessed 12 May 2023].
38. Adam, S. P., Alexandropoulos, S.-A. N., Pardalos, P. M., & Vrahatis, M. N. (2019). No free lunch theorem: A review. *Approximation and Optimization: Algorithms, Complexity and Applications*, pp. 57–82.
39. Scispace. (2023). <https://typeset.io/> [Online; accessed 12-May-2023].
40. Humata. (2023). <https://www.humata.ai/> [Online; accessed 12 May 2023].
41. Explainpaper. (2023). <https://www.explainpaper.com/> [Online; accessed 12-May-2023].
42. askyourpdf. (2023). <https://askyourpdf.com/> [Online; accessed 12-Jun-2023].
43. Gamma. (2023). <https://gamma.app/> [Online; accessed 4-June-2023].
44. Liu, V., & Chilton, L. B. (2022). Design guidelines for prompt engineering text-to-image generative models. In *Proceedings of the 2022 CHI conference on human factors in computing systems* (pp. 1–23).
45. White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt. [arXiv:2302.11382](https://arxiv.org/abs/2302.11382).
46. Oppenlaender, J. (2023). Prompt engineering for text-based generative art. [arXiv:2204.13988](https://arxiv.org/abs/2204.13988).
47. Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H., & Ba, J. (2022). Large language models are human-level prompt engineers. [arXiv:2211.01910](https://arxiv.org/abs/2211.01910).

48. Gill, S. S., Xu, M., Ottaviani, C., Patros, P., Bahsoon, R., Shaghaghi, A., Golec, M., Stankovski, V., Wu, H., Abraham, A., et al. (2022). Ai for next generation computing: Emerging trends and future directions. *Internet of Things*, 19, 100514.
49. Speechmatics. (2023). <https://portal.speechmatics.com/home/> [Online; accessed 12-May-2023].
50. Goossens, M., Mittelbach, F., Samarin, A., et al. (1994). *The LATEX companion* (Vol. 1). Addison-Wesley Reading.
51. Salvagno, M., Taccone, F. S., Gerli, A. G., et al. (2023). Can artificial intelligence help for scientific writing? *Critical Care*, 27(1), 1–5.
52. Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D., & Finn, C. (2023). Detectgpt: Zero-shot machine-generated text detection using probability curvature. [arXiv:2301.11305](https://arxiv.org/abs/2301.11305).
53. Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., & Tang, J. (2021). Gpt understands, too. [arXiv:2103.10385](https://arxiv.org/abs/2103.10385).
54. Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L., & Chen, W. (2021). What makes good in-context examples for gpt-3?. [arXiv:2101.06804](https://arxiv.org/abs/2101.06804).
55. Peng, B., Li, C., He, P., Galley, M., & Gao, J. (2023). Instruction tuning with gpt-4. [arXiv:2304.03277](https://arxiv.org/abs/2304.03277).
56. Zhou, J., Gandomi, A. H., Chen, F., & Holzinger, A. (2021). Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics*, 10(5), 593.
57. Guo, B., Zhang, X., Wang, Z., Jiang, M., Nie, J., Ding, Y., Yue, J., & Wu, Y. (2023). How close is chatgpt to human experts? Comparison corpus, evaluation, and detection. [arXiv:2301.07597](https://arxiv.org/abs/2301.07597).
58. Fu, J., Ng, S.-K., Jiang, Z., & Liu, P. (2023). Gptscore: Evaluate as you desire. [arXiv:2302.04166](https://arxiv.org/abs/2302.04166).
59. Betzalel, E., Penso, C., Navon, A., & Fetaya, E. (2022). A study on the evaluation of generative models. [arXiv:2206.10935](https://arxiv.org/abs/2206.10935).
60. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems* 30.
61. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. *Advances in Neural Information Processing Systems* 29.
62. Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International conference on machine learning* (pp. 8748–8763). PMLR.
63. You, H., Zhou, L., Xiao, B., Codella, N., Cheng, Y., Xu, R., Chang, S.-F., & Yuan, L., (2022) “Learning visual representation from modality-shared contrastive language-image pre-training. In *Computer Vision-ECCV. 17th European conference, Proceedings, Part XXVII* (pp. 69–87). Springer.
64. Campr, M., & Ježek, K. (2015). “Comparing semantic models for evaluating automatic document summarization. In *Text, Speech, and Dialogue: 18th International Conference, TSD, Proceedings 18* (pp. 252–260). Springer.
65. Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (pp. 311–318).
66. Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. [arXiv:1904.09675](https://arxiv.org/abs/1904.09675).
67. Banerjee, S., & Lavie, A. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization* (pp. 65–72).
68. Zingo, P. A., & Novocin, A. P. (2023). Qualitative analysis of synthetic computer network data using umap. In *Advances in information and communication: Proceedings of the 2023 future of information and communication conference (FICC)* (Vol. 2, pp. 849–861).

69. Zheng, S., Huang, J., & Chang, K. C.-C. (2023). Why does chatgpt fall short in answering questions faithfully?. [arXiv:2304.10513](https://arxiv.org/abs/2304.10513).
70. Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., A. Abid, Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. (2022). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. [arXiv:2206.04615](https://arxiv.org/abs/2206.04615).
71. Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., & Steinhardt, J. (2020). Measuring massive multitask language understanding. [arXiv:2009.03300](https://arxiv.org/abs/2009.03300).
72. Lin, S., Hilton, J., & Evans, O. (2021). Truthfulqa: Measuring how models mimic human falsehoods. [arXiv:2109.07958](https://arxiv.org/abs/2109.07958).
73. Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., et al. (2022). Holistic evaluation of language models. [arXiv:2211.09110](https://arxiv.org/abs/2211.09110).
74. Mahowald, K., Ivanova, A. A., Blank, I. A., Kanwisher, N., Tenenbaum, J. B., & Fedorenko, E. (2023). Dissociating language and thought in large language models: A cognitive perspective. [arXiv:2301.06627](https://arxiv.org/abs/2301.06627).
75. Boiko, D. A., MacKnight, R., & Gomes, G. (2023). Emergent autonomous scientific research capabilities of large language models. [arXiv:2304.05332](https://arxiv.org/abs/2304.05332).
76. Rajkumar, N., Li, R., & Bahdanau, D. (2022). Evaluating the text-to-sql capabilities of large language models. [arXiv:2204.00498](https://arxiv.org/abs/2204.00498).
77. Armengol-Estabé, J., Bonet, O. d. G., & Melero, M. (2021). On the multilingual capabilities of very large-scale english language models. [arXiv:2108.13349](https://arxiv.org/abs/2108.13349).
78. Frieder, S., Pinchetti, L., Griffiths, R.-R., Salvatori, T., Lukasiewicz, T., Petersen, P. C., Chevallier, A., & Berner, J. (2023). Mathematical capabilities of chatgpt. [arXiv:2301.13867](https://arxiv.org/abs/2301.13867).
79. Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., Lovenia, H., Ji, Z., Yu, T., Chung, W., et al. (2023). A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. [arXiv:2302.04023](https://arxiv.org/abs/2302.04023).
80. Zhong, Q., Ding, L., Liu, J., Du, B., & Tao, D. (2023). Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert. [arXiv:2302.10198](https://arxiv.org/abs/2302.10198).
81. Sobania, D., Briesch, M., Hanna, C., & Petke, J. (2023). An analysis of the automatic bug fixing performance of chatgpt. [arXiv:2301.08653](https://arxiv.org/abs/2301.08653).
82. Wang, J., Hu, X., Hou, W., Chen, H., Zheng, R., Wang, Y., Yang, L., Huang, H., Ye, W., & Geng, X., et al. (2023). On the robustness of chatgpt: An adversarial and out-of-distribution perspective. [arXiv:2302.12095](https://arxiv.org/abs/2302.12095).
83. Borji, A. (2023). A categorical archive of chatgpt failures. [arXiv:2302.03494](https://arxiv.org/abs/2302.03494).
84. Zhuo, T. Y., Huang, Y., Chen, C., & Xing, Z. (2023). Exploring ai ethics of chatgpt: A diagnostic analysis. [arXiv:2301.12867](https://arxiv.org/abs/2301.12867).
85. Wang, C., Liu, P., & Zhang, Y. (2021). Can generative pre-trained language models serve as knowledge bases for closed-book qa?. [arXiv:2106.01561](https://arxiv.org/abs/2106.01561).
86. Sorensen, T., Robinson, J., Rytting, C. M., Shaw, A. G., Rogers, K. J., Delorey, A. P., Khalil, M., Fulda, N., & Wingate, D. (2022). An information-theoretic approach to prompt engineering without ground truth labels. [arXiv:2203.11364](https://arxiv.org/abs/2203.11364).
87. Jiang, Z., Araki, J., Ding, H., & Neubig, G. (2021). How can we know when language models know? On the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9, 962–977.
88. Roberts, A., Raffel, C., & Shazeer, N. (2020). How much knowledge can you pack into the parameters of a language model?. [arXiv:2002.08910](https://arxiv.org/abs/2002.08910).
89. Alkhaldi, T., Chu, C., & Kurohashi, S. (2022). A peek into the memory of t5: Investigating the factual knowledge memory in a closed-book qa setting and finding responsible parts. *Journal of Natural Language Processing*, 29(3), 762–784.
90. Zhu, F., Lei, W., Wang, C., Zheng, J., Poria, S., & Chua, T.-S. (2021). Retrieving and reading: A comprehensive survey on open-domain question answering. [arXiv:2101.00774](https://arxiv.org/abs/2101.00774).
91. Dong, X., Lu, J., Wang, J., & Caverlee, J. (2022)). Closed-book question generation via contrastive learning. [arXiv:2210.06781](https://arxiv.org/abs/2210.06781).

92. Li, K., Patel, O., Viégas, F., Pfister, H., & Wattenberg, M. (2023). Inference-time intervention: Eliciting truthful answers from a language model. [arXiv:2306.03341](https://arxiv.org/abs/2306.03341).
93. Altmann, J., & Gubrud, M. (2004). Anticipating military nanotechnology. *IEEE Technology and Society Magazine*, 23(4), 33–40.
94. Li, Y., Lin, Z., Zhang, S., Fu, Q., Chen, B., Lou, J.-G., & Chen, W. (2022). On the advance of making language models better reasoners. [arXiv:2206.02336](https://arxiv.org/abs/2206.02336).
95. Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., & Zhou, D. (2022). Self-consistency improves chain of thought reasoning in language models. [arXiv:2203.11171](https://arxiv.org/abs/2203.11171).
96. Došilović, F. K., Brčić, M., & Hlupić, N. (2018). Explainable artificial intelligence: A survey. In *41st International convention on information and communication technology, electronics and microelectronics (MIPRO)* (pp. 0210–0215). IEEE.
97. Sun, J., Liao, Q. V., Muller, M., Agarwal, M., Houde, S., Talamadupula, K., & Weisz, J. D. (2022). Investigating explainability of generative ai for code through scenario-based design. In *27th International conference on intelligent user interfaces* (pp. 212–228).
98. Xu, M., Du, H., Niyato, D., Kang, J., Xiong, Z., Mao, S., Han, Z., Jamalipour, A., Kim, D. I., Leung, V., et al. (2023). Unleashing the power of edge-cloud generative ai in mobile networks: A survey of aigc services. [arXiv:2303.16129](https://arxiv.org/abs/2303.16129).
99. Bozkurt, A., Xiao, J., Lambert, S., Pazurek, A., Crompton, H., Koseoglu, S., Farrow, R., Bond, M., Nerantzi, C., Honeychurch, S., et al. (2023). Speculative futures on chatgpt and generative artificial intelligence (ai): A collective reflection from the educational landscape. *Asian Journal of Distance Education* 18(1).
100. Zhang, Z., Zhang, A., Li, M., & Smola, A. (2022). Automatic chain of thought prompting in large language models. [arXiv:2210.03493](https://arxiv.org/abs/2210.03493).
101. Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Bousquet, O., Le, Q., & Chi, E. (2022). Least-to-most prompting enables complex reasoning in large language models. [arXiv:2205.10625](https://arxiv.org/abs/2205.10625).
102. Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. [arXiv:2205.11916](https://arxiv.org/abs/2205.11916).
103. Xu, S., Pang, L., Shen, H., Cheng, X., & Chua, T.-S. (2023). Search-in-the-chain: Towards the accurate, credible and traceable content generation for complex knowledge-intensive tasks. [arXiv:2304.14732](https://arxiv.org/abs/2304.14732).
104. Pitis, S., Zhang, M. R., Wang, A., & Ba, J. (2023). Boosted prompt ensembles for large language models. [arXiv:2304.05970](https://arxiv.org/abs/2304.05970).
105. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., & Zhou, D. (2022). Chain of thought prompting elicits reasoning in large language models. [arXiv:2201.11903](https://arxiv.org/abs/2201.11903).
106. Andrade, F. (2023). Noteable: The chatgpt plugin that automates data analysis. <https://artificialcorner.com/noteable-the-chatgpt-plugin-that-automates-data-analysis> [Online; accessed 12-May-2023].

Dr. Abraham Itzhak Weinberg has spent over 30 years in the fields of software and information systems. He had served for six years in the IAF (Israeli Air Force) and retired as a Captain. In recent years, he has managed a BI (business intelligence) unit and data warehousing projects and has consulted data science projects as well as projects integrating big data and cybersecurity. His academic background consists of a BSc in industrial engineering and management as well as computer science, and an MSc in industrial engineering. Dr. Weinberg obtained his PhD in Software and Information Systems Engineering. He also completed his post-doctorate in the United Kingdom. As a lecturer, he had the opportunity to teach BI courses, facilitate seminars, and contribute to the publication of papers in esteemed journals and conferences. His main research interests are focused on Artificial intelligence, Generative AI, data science, Explainability (XAI), big data, and cyber security.

Creating Ad Campaigns Using Generative AI



Ahmet Bulut and Bariş Arslan

Abstract Search campaigns consist of ad groups. An ad group contains a related set of keywords and ads. During an online campaign, search advertisers experiment with different marketing messages such as subtle vs. strong being used in ad copies, with different keywords from broad match to exact match while targeting online users, and with different landing pages to target information seekers vs. product buyers. Generating new ads and keywords for this experimental endeavor becomes essential in effective search campaign management. In this work, we discuss the role of generative AI in creating and managing search ad campaigns programmatically.

1 Introduction

Generative AI has been successfully used in generating text content such as stories, poems, social media posts, and literature reviews [29]. It is applicable in search advertising as well [4]. Search advertising is a specific type of online marketing where text based ads are shown to a search engine user in case her search query matches with a keyword found in a search campaign. A search campaign consists of multiple ad groups. Each ad group contains multiple ads and keywords that are related to each other. Figure 1 shows the structure of a search campaign. An ad contains a marketing message and a link to the landing page of the advertised product. One of the ads in the first ad group of the second campaign, i.e., *Adgroup (2,1)*, for advertising online sql courses is given below:

learn sql | manage your database
Learn the industry best practices.
<http://www.learn.com/courses/sql>.

A. Bulut (✉) · B. Arslan

Acibadem University, Department of Computer Science, 34752 Ataşehir, İstanbul, Turkey
e-mail: ahmet.bulut@acibadem.edu.tr

B. Arslan
e-mail: baris.arslan@acibadem.edu.tr

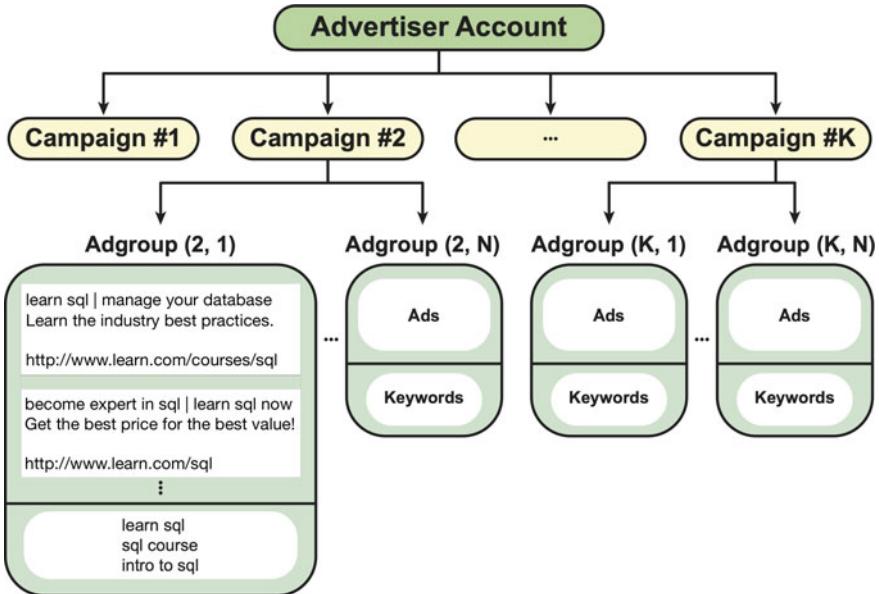


Fig. 1 Advertisers create campaigns, which consist of multiple ad groups. Each ad group contains a set of related keywords and ads

In the above example, `learn sql` and `manage your database` are the headlines of the ad. The phrase `Learn the industry best practices` is the description. Finally, the ad has a link to the webpage of the online course. The keywords chosen for targeting this ad are `learn sql`, `sql course`, and `intro to sql`. If a query posed by a search engine user matches with any of these keywords partially or exactly, this ad will be displayed to that user along with other relevant search results. In general, a text ad in Google has up-to three headlines, each containing 30 characters at most. Headlines are separated by a pipe symbol, i.e. `|`. In addition, the ad has up-to two descriptions, each containing 90 characters at most.

Various studies in the search advertising literature address either keyword generation or ad generation individually [21, 25]. However, a search campaign consists of both ads and keywords. Therefore, it is not one or the other, but it is both that one needs to address.

2 Foundations

Theoretically, a learnt language model processes a given text as input and produces a corresponding text as output. The produced output could be viewed as an adaptation of the given input. As such, the language modeling would be a suitable approach for text generation in search advertising as shown in Fig. 2. In case of keyword

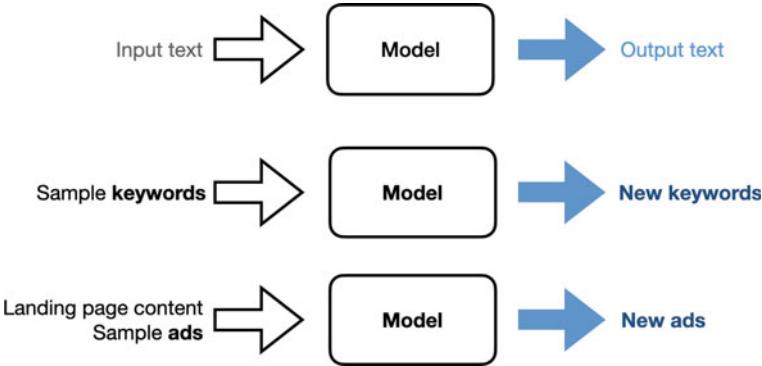


Fig. 2 The learnt language model processes the input text, and produces the output text. In both of our text generation tasks, the produced output could be viewed as an adaptation of the given input

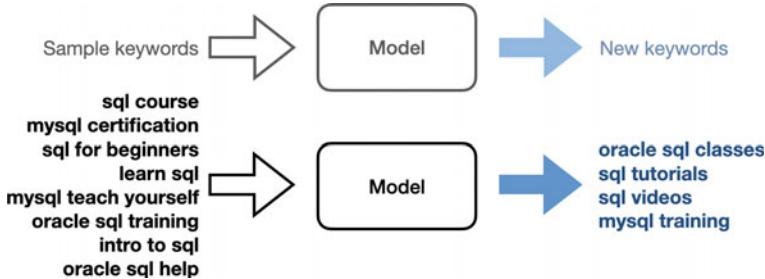


Fig. 3 The existing keywords are used as input, and the output is a new keyword

generation, existing keywords are used as input, and the output is a new keyword as shown in Fig. 3. In case of ad generation, existing ads and landing page content are used as input, and the output is a new ad as shown in Fig. 4.

A generative model estimates from a given a sequence of words the probability of the next word among all possible words. The estimates are higher for words that appear more frequently in the training data at a given position under consideration.

RNN A Recurrent Neural Network (RNN) is a simple model to use for generating text [19]. An RNN processes input text one word at a time. Its output is fed recursively as input to the model itself in order to capture any temporal dependencies that may be present in the data. RNNs were shown to work well in modeling user preference [6, 9]. An RNN retains information about the previous tokens in a given sequence, and it pays equal attention to all tokens. As the length of the sequence increases, it could forget important information due to its limited memory.

LSTM A basic RNN can remember the next word, but one may need to know more than just the next word. A concise summary of the context should be retained in order to generate novel output for creative copy-writing. A long short-term memory network (LSTM) has a local memory for persisting important information [7]. It captures what information to forget and what to retain at every time step.

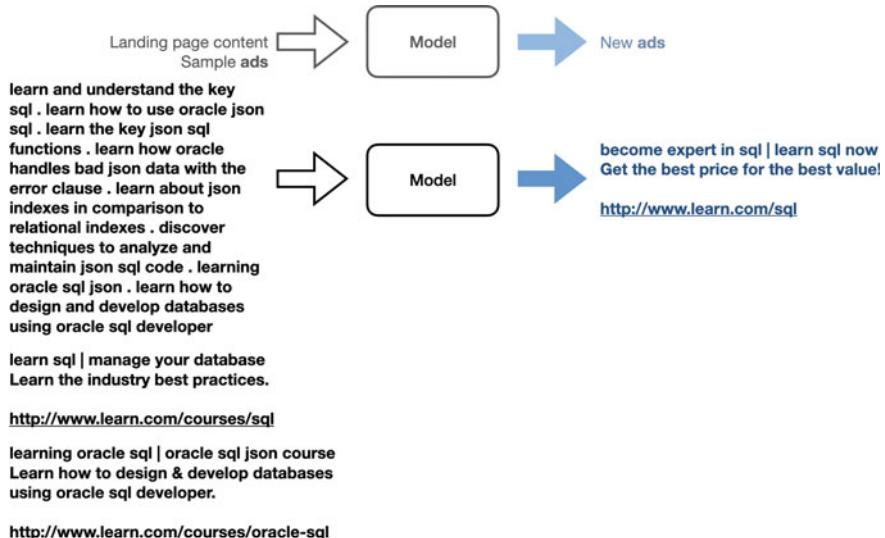


Fig. 4 The landing page content and the existing ads are used as input, and the output is a new ad

Transformer The transformer architecture proposed by Vaswani et al. paved the way for advancing generative models. It underpinned the importance of attention mechanism [24]. Bahdanau et al. had discovered that assigning weighted attention to each input would yield more accurate results [1]. The transformer has shown exemplary performance in many applications ranging from machine translation and text generation to abstractive summarization [12]. Though RNNs process text sequentially, a transformer does not process input sequentially, and hence could be parallelized on GPUs. A transformer consists of multiple encoders and decoders. The input is fed into the first encoder in the encoder stack. After being shifted to the right, the output is fed into the first decoder in the decoder stack. The architecture uses multi-head attention in three different places: the self-attention layers in the encoders, the self-attention layers in the decoders, and the “encoder-decoder attention” layers, which allows the decoder to attend over the input sequence. This particular wiring is adopted to mimic the typical encoder-decoder attention mechanisms found in basic sequence-to-sequence models.

GPT A generative pre-training (GPT) model is based on the transformer architecture. It is trained on much more data, e.g., public web pages, has a very large number of parameters, and contains self-attention layers in just the decoders [17]. For instance, a relatively smaller version of GPT called GPT-2 Small contains 117M parameters, and has a word vocabulary size of 50K. The model can be used in question answering, text summarization, and language translation simply by providing appropriate task names called prompts [3].

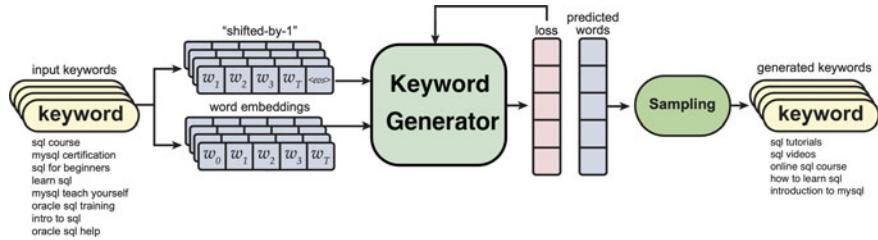


Fig. 5 The flowchart illustrates the various steps involved in generating keywords

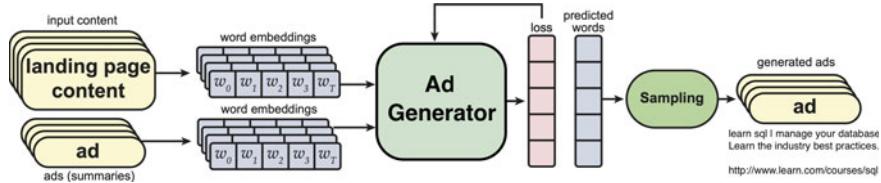


Fig. 6 The flowchart illustrates the various steps involved in generating ads

3 Methodology

In this section, we describe the mechanics of generative models, which are suitable for creating ad campaigns programmatically. Most of the models that performed well in practice across a wide range of applications all have a similar configuration. They are based on the encoder-decoder network architecture. They vary by the number of encoders and decoders used, and by the particular application of the attention mechanism.

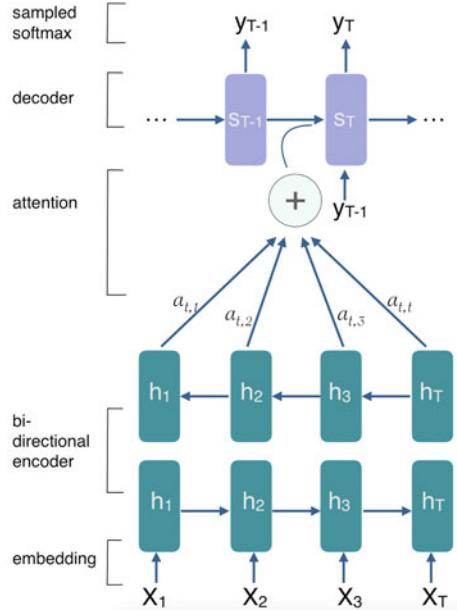
3.1 Preliminaries

Input text is tokenized first. Then, the resulting tokens are vectorized using text transformation. There are various methods used in practice for this purpose including Word2Vec [13], GloVe [15], and fastText [2]. The goal is to embed words into a vector space of a fixed dimension such that each word is represented as a unique vector of its meaningful contexts.

In order to train a model for generating text, the output loss at a given time step t should be minimized with respect to the output at time step $t - 1$. For a generated text of length T , the cross-entropy loss is computed as follows:

$$L = -\frac{1}{T} \sum_{t=0}^T \sum_{j=1}^{|V|} P(w_{j,t+1}) \log \hat{P}(w_{j,t+1} | w_{j,t}) \quad (1)$$

Fig. 7 The illustration of an encoder-decoder network architecture for summarizing text input: X_t s are sequences from the input document, h_t s are hidden states in the encoder phase, a_t s are part of the attention mechanism, S_t s are the decoder states. Y_t s are the output sequences that summarize a given input document



where w represents a word, $P(w_{j,t+1})$ denotes the probability of the true word, and $|V|$ denotes the cardinality of the word vocabulary.

Initially, a seed word has to be provided for the model to start generating text. For instance, one seed would suffice when generating short text. The initial seed could be selected by random weighted sampling. A frequency distribution could be obtained using the first token of each input text. During text generation, one or more seeds could be sampled from this distribution. This method would result in a distribution of generated text that resembles the original distribution, but at the same time enables generating text that is novel, i.e., not seen before as it is.

Figures 5 and 6 show the various steps involved in generating the core elements of any campaign, i.e., ads and keywords.

3.2 Encoder-Decoder Architecture

An encoder-decoder architecture is shown in Fig. 7. Using this network, the input sequence is mapped to the output sequence using word vectors. Word embedding is used in this example for transforming text. The encoder reads input sequences of varying length and encodes them into vectors. These vectors retain the context information present in the input. The decoder emits the final word predictions using the output vectors of the encoder. Within the cells of each encoder and decoder, an LSTM or an RNN network could be used.

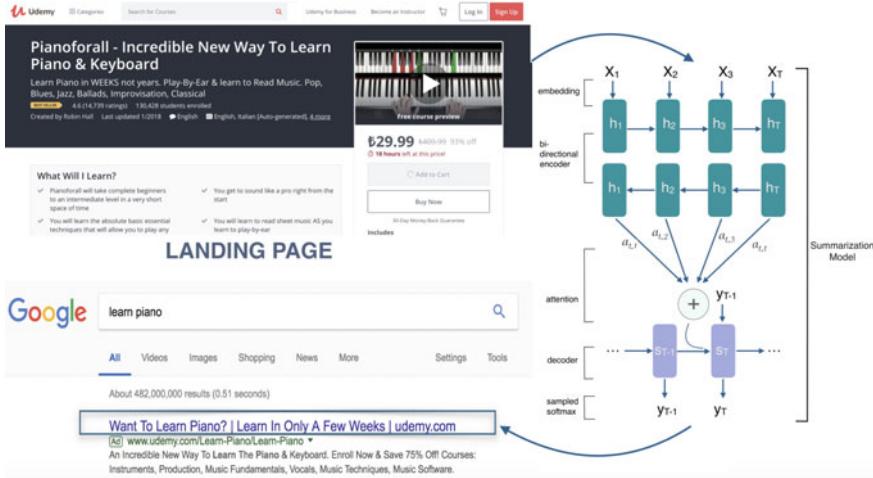


Fig. 8 The summarization of a landing page into a text advertisement using generative AI

A bidirectional LSTM or a bidirectional RNN as in this particular example could also be used in the encoding stage in order to retain information present in the past as well as the future since the reversal of the input in a given stage allows the network to retain information that occurs in the “future” [20]. For soft-alignment between each input word and output word, the attention mechanism is added into the model. At the final step, a softmax sampling is applied in order to compute the final probabilities over the target vocabulary.

An advertisement could be generated from landing page content, which typically is much longer in size. Therefore, such a task is usually treated as text summarization. Figure 8 illustrates the use of the aforementioned model in a pipeline for generating an ad from landing page content. Specifically, Hughes et al. used two encoder-decoder RNNs in parallel, one for generating the headlines of an ad, and the other for generating the descriptions of an ad. On a dataset containing landing page to ad pairs, their model learnt the association between the two [8]. Çoğalmış and Bulut used a model similar to the one shown in Fig. 7 in order to create ads from landing page content [5].

Generative adversarial networks (GAN) were used for novel text generation [32]. The performance of a text generator network [22] can be enhanced by using a generative adversarial training approach as illustrated in Fig. 9. For this purpose, a discriminator, e.g., a self-critic adversarial reinforce network [27], could be used. The discriminator takes an input word sequence from the generator and emits a score for the input sequence by considering the actual word sequence. When the discriminator encounters a grammatically incorrect structure such as repetitive words coming from the generator, it penalizes the generator by assigning such sequences relatively lower scores.

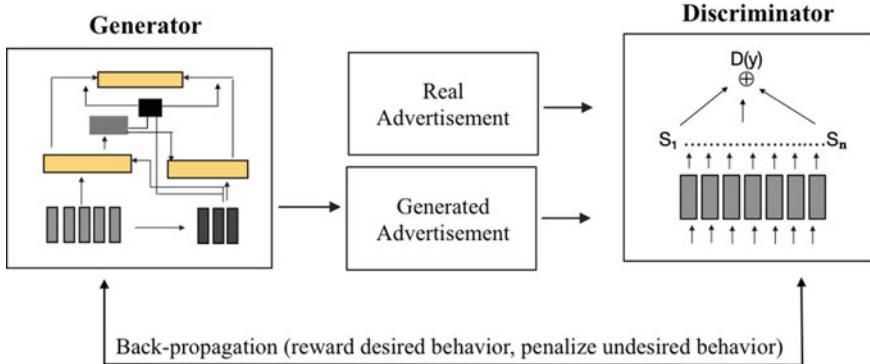


Fig. 9 Reinforcing a text generator with a discriminator for steering the generator in the desired direction by penalizing any undesired behavior

Using reinforcement learning (RL), Wang et al. showed how the performance of pre-trained models could be improved further in generating high-quality text ads [26]. In a similar manner, Terzioğlu et al. studied the generation of ads in the context of RL. They proposed a GAN model for this purpose where the generator is an encoder-decoder type bi-directional RNN with attention, and the discriminator is a single-layer uni-directional LSTM [23]. Their experimental results showed that the reinforced generative model performed well in creating ad creatives programmatically using the descriptive information found in the landing pages. The specific contribution of reinforcement in model performance need to be studied in detail and should be quantified.

Using a GPT based model, Bulut and Mahmoud showed that new campaign keywords could be generated easily by using an appropriate task prompt. The task token they used is “*keyword*:”, which is inserted into the beginning of all keyword instances before being fed into the generative model [4]. According to their empirical results, the keywords generated by a specific GPT-2 model were expected to get a higher number of unique user clicks and conversions compared to the keywords generated by other more standard models, e.g., LSTM. Compared to a typical transformer based model, a GPT based model has a non-standard application of the attention mechanism, and it includes a pre-training stage. Pre-trained language models based on transformers, which are trained on large corpora, are known to avoid over-fitting and to represent the underlying language well [16]. Hence, training a language model on a very large dataset, and then fine-tuning it for a specific language task at hand is a viable approach. This method is known as transfer learning in the literature. For instance, BART [10], T5 [18], and PEGASUS [31] are representative models in this regard. The web portal by Hugging Face¹ hosts a collection of such pre-trained transformers libraries [28]. In particular, T5 and PEGASUS models were trained on 750 GB of English-language text from the public Common Crawl web scrape while

¹ <https://huggingface.co/models>.

the BART model was trained on the CNN/Daily Mail dataset, which contains roughly 300K unique news articles from CNN and the Daily Mail. For a domain-specific task, they are customized with an additional training step using a domain-specific dataset. The effect of pre-training in the field performance of a generative model is worth investigating further.

4 Empirical Results

In this section, we present the details of an empirical study that shows how the performance of representative models change with respect to key algorithmic choices: the particular use of the attention mechanism, the impact of reinforcement, and the effect of pre-training.

4.1 Datasets

Table 1 shows sample rows from the ads dataset. It contains 4795 rows. Each row is a pair of landing page content and the corresponding ad. The landing page title is

Table 1 Sample rows from the ads dataset

Landing page text	Ad creative
create your own ios apps . apply for ios developer jobs . choose the best design pattern for your app . monetize your skills . upload your own ios apps to the app store . the complete ios swift + objective c developer course . this course will teach you both the swift and objective c programming languages and how to build ios mobile apps	the complete ios swift objective c developer course create ios apps . learn the swift and objective c programming . learn how to build ios mobile apps
create and write advanced functions . create charts . and buttons sort and filter with regular and advanced filters . build pivot tables and calculated columns . use randomness for model prediction . learning microsoft excel : introduction to mastery . using excel to build advanced functions	learning microsoft excel introduction to mastery excel . using excel to build advanced functions and financial models . build pivot tables and use randomness for model prediction
understand the purpose of webpack in a modern web app . build custom boilerplate projects to serve es2015 javascript . deploy heroku , and more enhance the performance of web apps by leveraging webpack's ecosystem of plugins . enhance code organization through the use of es2015 js modules webpack 2 : the complete developer's guide . master webpack 2 as you deploy web apps supported by babel , code splitting , and es2015 modules	webpack guide for beginner master webpack 2 with web apps . as you deploy web apps supported by babel , code splitting , and es2015 modules . deploy webpack based projects to aws , heroku , and more

Table 2 The effect of attention mechanism on model performance

Configuration	ROUGE-1	ROUGE-2	ROUGE-L	BLEU
Model without attention	22.6	5.8	21.5	4.5
Model with attention	56.9	42.5	55.9	32.8

^a The base model is a simple RNN. The only difference is whether or not an attention layer exists in the network architecture

also included in the landing page content. The ads adhere to Google’s ad format. The keywords dataset contains 52K keywords that reside in 260 campaigns. There is a row of data per keyword, including keyword text as well as campaign and ad group identifiers.

4.2 Evaluation Metrics

The generated text can be evaluated by checking whether it is grammatically correct and whether the sentences are well-structured. These are measurements for its syntactical quality. The content-based metrics measure text quality by comparing the words in the generated sentences with the words in the human-generated sentences. The Bilingual Evaluation Understudy (BLEU) is a widely used metric for measuring the quality of generated text [14]. It is defined as the fraction of n -grams in the generated text that also appear in the ground-truth text. Another widely used metric in this space is Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [11]. ROUGE- n represents the ratio of n -grams found in the ground-truth text that also appear in the generated text. The longest overlapping subsequence between the generated text and the ground-truth text is given by ROUGE- L . When used together, BLEU and ROUGE shed light onto textual coherence and syntactic text quality. Yuan et al. studied the classification and the use of persuasive tactics in ad text, and predicted the promotional effectiveness of a given ad [30]. Such quantitative metrics are useful for the performance evaluation of a generative model in addition to the syntactic text quality scores.

Online advertisers assess the performance of a keyword or an ad by the rate at which it gets user clicks (click-through), and by the rate at which those users convert post clicks. Both clicks and conversions are equally important, and they represent user response. The target response depends on the goal of the marketing campaign. For raising brand awareness, the target response is user clicks; for direct sales and profit, the target response is user conversions. The total number of clicks that a keyword gets represents its click appeal to users. In the same manner, the total number of conversions that a keyword gets represents its sales potential.

Table 3 The effect of reinforcement learning on model performance

Configuration	ROUGE-1	ROUGE-2	ROUGE-L
Model without reinforcement	34.0	19.0	34.0
Model with reinforcement	58.3	33.8	56.5

^a The base model is an encoder-decoder type model. Bi-directional RNNs are used in the encoder and decoder

4.3 The Effect of Attention

We performed an experiment for testing the effect of attention mechanism while generating ads. We measured the ROUGE and BLEU performance of an RNN with attention, and compared it with the performance of RNN without an attention layer. As shown in Table 2, the application of attention improved model performance dramatically.

The proper application of the attention mechanism elevates the performance of a generative model.

4.4 The Effect of Reinforcement

We performed an experiment for testing the impact of reinforcement while generating ads. We measured the ROUGE performance of an encoder-decoder type model with reinforcement and without any reinforcement. As shown in Table 3, the application of reinforcement improved model performance.

Reinforcing a generator with a discriminator to steer the generator in the desired direction by penalizing its undesired behavior such as excessive repetitions and grammatical errors elevates its performance.

4.5 The Effect of Pre-training

For a given set of keywords, Google's keyword planner service provides a forecast of the expected volume of unique user clicks and conversions the set could generate. The forecast is a valuable measurement for comparing different keyword generation models. We performed an experiment for testing the impact of pre-training while

Table 4 The effect of pre-training over large and diverse data on model performance

Configuration	Unique user clicks	Unique user conversions
Model without pre-training	10	1
Model with pre-training	720	55

^a The model with pre-training is a GPT-2 model. The model without any pre-training is an LSTM model

Table 5 The effect of pre-training over a large text corpus on model performance

Configuration	ROUGE-1	ROUGE-2	ROUGE-L
Model without pre-training	34.0	19.0	34.0
Model with pre-training	48.4	33.9	41.9

^a The model without pre-training is an encoder-decoder type model with bi-directional RNNs being used in the encoder and decoder. The model with pre-training is a “transformer” encoder-decoder type model with a bidirectional encoder and an autoregressive decoder

generating keywords. We compared the outreach performance of a model, which is trained on the keywords data only, with a model that has a pre-training stage on public data before being customized on the keywords data. The pre-training enhances the model’s exploration capacity as shown by the higher number of unique user clicks and conversions in Table 4.

Additionally, we performed another experiment for testing the impact of pre-training while generating ads. We measured the ROUGE performance of a pre-trained language model based on transformers, which also has an additional task-specific training stage, and compared it with a similar model that shares the basic network architecture but lacks a particular stage of pre-training on a large text corpus. As shown in Table 5, the pre-training improved model performance.

Pre-training on “out-of-context” data from various domains enhances a generative model’s exploration capacity while generating “within-context” data for a target domain.

5 Conclusion

We studied how well the encoder-decoder type generative models perform in search advertising. Specifically, we investigated how the performance of representative models change with respect to key algorithmic choices including the particular use of the attention mechanism, the impact of reinforcement learning, and the effect of an optional pre-training stage. We found out that all of the three choices affect the final model performance greatly; the lift-up in performance was up to an order of magnitude.

Acknowledgements The authors thank Dr. Kevser Nur Çoğalmış from İstanbul Sabahattin Zaim University, Süimeyra Terzioğlu from Kuveyt Türk R&D Center, Abdelrahman Mahmoud and Fahed Şabellioglu from Oredata Turkey. Their efforts laid the foundation for this overarching work.

References

1. Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *3rd International conference on learning representations, ICLR 2015*.
2. Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5, 135–146.
3. Bommasani, R., et al. (2021). On the opportunities and risks of foundation models. CoRR [abs/2108.07258](https://arxiv.org/abs/2108.07258). <https://arxiv.org/abs/2108.07258>
4. Bulut, A., & Mahmoud, A. (2023). Generating campaign ads and keywords for programmatic advertising. *IEEE Access*, 11, 43557–43565. <https://doi.org/10.1109/ACCESS.2023.3269505>
5. Çoğalmış, K. N., & Bulut, A. (2022). Generating ad creatives using deep learning for search advertising. *Turkish Journal of Electrical Engineering and Computer Sciences*, 30(5), 1882–1896.
6. Chambua, J., Niu, Z., & Zhu, Y. (2019). User preferences prediction approach based on embedded deep summaries. *Expert Systems with Applications*, 132, 87–98.
7. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-term Memory. *Neural Computation*, 9(8), 1735–1780.
8. Hughes, J. W., Chang, K. H., & Zhang, R. (2019). Generating better search engine text advertisements with deep reinforcement learning. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '19* (pp. 2269–2277). Association for Computing Machinery.
9. Koehn, D., Lessmann, S., & Schaal, M. (2020). Predicting online shopping behaviour from clickstream data using deep learning. *Expert Systems with Applications* 150
10. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. CoRR [abs/1910.13461](https://arxiv.org/abs/1910.13461). [http://arxiv.org/abs/1910.13461](https://arxiv.org/abs/1910.13461).
11. Lin, C. Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74–81). Association for Computational Linguistics.
12. Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., & Shazeer, N. (2018). Generating wikipedia by summarizing long sequences. CoRR [abs/1801.10198](https://arxiv.org/abs/1801.10198). [http://arxiv.org/abs/1801.10198](https://arxiv.org/abs/1801.10198).
13. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2–4 May 2013). Efficient estimation of word representations in vector space. In *1st International conference on learning representations, ICLR 2013 Workshop Track Proceedings*.
14. Papineni, K., Roukos, S., Ward, T., & Zhu, W.J. (2002). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 311–318). Association for Computational Linguistics.
15. Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543).
16. Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., & Huang, X. (2020). Pre-trained models for natural language processing: A survey. CoRR [abs/2003.08271](https://arxiv.org/abs/2003.08271). <https://arxiv.org/abs/2003.08271>.
17. Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

18. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1), 5485–5551.
19. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536.
20. Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11), 2673–2681.
21. Schwaighofer, A., Candela, J.Q.n., Borchert, T., Graepel, T., & Herbrich, R. (2009). Scalable clustering and keyword suggestion for online advertisements. In *Proceedings of the 3rd international workshop on data mining and audience intelligence for advertising, ADKDD '09* (pp. 27–36). <https://doi.org/10.1145/1592748.1592753>
22. See, A., Liu, P.J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. CoRR **abs/1704.04368**. <http://arxiv.org/abs/1704.04368>.
23. Terzioğlu, S., Çoğalmış, K.N., & Bulut, A. (2022). Ad creative generation using reinforced generative adversarial network. *Electronic Commerce Research*. <https://doi.org/10.1007/s10660-022-09564-6>
24. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems (NIPS)*, pp. 5998–6008.
25. Vempati, S., Malayil, K.T., V. S., & R, S. (2019). Enabling hyper-personalisation: Automated ad creative generation and ranking for fashion e-commerce. CoRR **abs/1908.10139**.
26. Wang, X., Gu, X., Cao, J., Zhao, Z., Yan, Y., Middha, B., & Xie, X. (2021). Reinforcing pretrained models for generating attractive text advertisements. In *ACM SIGKDD international conference on knowledge discovery and data mining (ACM SIGKDD)* (pp. 3697–3707).
27. Wang, Y., & Lee, H.Y. (2018). Learning to encode text as human-readable summaries using generative adversarial networks. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 4187–4195). <https://doi.org/10.18653/v1/D18-1451>.
28. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., & Rush, A. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: System demonstrations* (pp. 38–45). <https://doi.org/10.18653/v1/2020.emnlp-demos.6>.
29. Yang, Z., Bai, H., Luo, Z., Xu, Y., Pang, W., Wang, Y., Yuan, Y., & Yuan, Y. (2023). Pacanet: A study on cyclegan with transfer learning for diversifying fused chinese painting and calligraphy. In arXiv preprint **arXiv:2301.13082**, submitted to IJCAI 2023.
30. Yuan, Y., Xu, F., Cao, H., Zhang, G., Hui, P., Li, Y., & Jin, D. (2021). Persuade to click: Context-aware persuasion model for online textual advertisement. *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1. <https://doi.org/10.1109/TKDE.2021.3110724>.
31. Zhang, J., Zhao, Y., Saleh, M., & Liu, P. J. (2019). PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. CoRR **abs/1912.08777**. <http://arxiv.org/abs/1912.08777>
32. Zhang, Y., Gan, Z., & Carin, L. (2016). Generating text via adversarial training. In *NIPS workshop on adversarial training*.

Unlocking the Potential of Generative Artificial Intelligence in Drug Discovery



Virgilio Romanelli, Carmen Cerchia, and Antonio Lavecchia

Abstract Deep generative models have been widely employed across diverse fields, ranging from image and video analysis to natural language processing. In combination with the increasing computational power and abundant data resources available in the public domain, generative models have made significant advancements into the area of drug discovery and development. In particular, generative models are being extensively explored for de novo design of novel molecules, endowed with desirable physicochemical properties or biological activity, thereby accelerating the hit discovery phase by more rapidly sampling the chemical space of drug-like compounds. However, despite their considerable potential, these methods do have limitations that warrant consideration. For instance, they tend to generate compounds that may exhibit chemical instability, pose challenges in synthesis, or bear resemblance to existing drugs, thereby raising concerns regarding patentability. Furthermore, the experimental validation of the generated molecules through exemplary case studies remains limited. This chapter focuses on the application of generative models in de novo drug design. Firstly, we provide a brief introduction to commonly used generative models, such as recurrent neural networks, autoencoders, generative adversarial networks, as well as transfer learning and reinforcement learning techniques. Secondly, we conduct a comprehensive review of the latest developments in utilizing various generative models for drug discovery. This includes an analysis of benchmarks, metrics, and performance evaluation methods through the examination of diverse case studies. Finally, we shed light on the challenges associated with generative methods and discuss future directions in this dynamic and rapidly evolving field.

Keywords De novo design · Generative models · Molecular representation · Artificial intelligence · Molecular design · Drug Discovery

V. Romanelli · C. Cerchia · A. Lavecchia (✉)

Department of Pharmacy, “Drug Discovery” Laboratory, University of Naples “Federico II”,
Via D. Montesano 49, 80131 Napoli, Italy
e-mail: antonio.lavecchia@unina.it

1 Introduction

The process from initial discovery to market approval of a new medicine is commonly recognized in the world of pharmaceutical research and development for its lengthy timeframe, which often lasts for more than a decade, complex nature, and significant financial commitments [1]. It involves sequential steps such as target discovery and validation, hit identification, lead optimization, preclinical and clinical research, regulatory approval, and post-marketing surveillance. Despite concerted efforts, the attrition rate of drug candidates remains high because of inherent safety risks, inadequate efficacy against targeted diseases, stringent regulatory requirements [2], and traditional methodologies like high throughput screening (HTS), a process that screens thousands to millions of compounds, or combinatorial chemistry encounter substantial challenges. To overcome these challenges, computational approaches have emerged as significant tools in the field of pharmaceutical research, integrating and improving the efficiency of traditional methodologies. By leveraging computing capacity, these techniques can model, simulate, and evaluate molecular interactions and processes. In contrast to traditional screening, virtual screening enables the quick screening of sizable compound libraries, assisting in the discovery of prospective drug candidates with the desired interactions and characteristics [3]. Approaches like docking and molecular dynamics facilitate the evaluation of the binding affinity between small compounds and target proteins.

De novo drug design is a highly successful computational approach that involves generating new potential ligands from scratch, avoiding explicit enumeration of the vast space of possible structures [4]. However, de novo design faces the challenge of the vast “chemical universe”, estimated to contain up to 10^{60} small molecular entities [5], making extensive enumeration impractical. Recent advances in artificial intelligence (AI) and deep learning have greatly benefited de novo design [6].

In recent years, deep generative models have emerged as a captivating field of research, capturing increasing attention and sparking widespread fascination. These models strive to comprehend the underlying probability distribution of training data, extracting representative features that encapsulate its essence. By constructing a low-dimensional continuous representation, they equip themselves with the remarkable ability to generate entirely new data points by sampling from the acquired data distribution [7]. The implications of generative models have reverberated across a multitude of domains, delivering exceptional outcomes in the realm of image generation [8], textual synthesis [9], speech reproduction [10], and even music composition [11]. The development of generative models has also propelled innovative ideas and fresh perspectives within the challenging realm of drug design. Unlike traditional approaches that rely heavily on human expertise, generative models leverage recent advancements in deep learning to tackle the inverse molecular design problem: determining the set of molecules that possess a desired set of properties. By establishing a mapping function between properties and molecular structures, generative models can swiftly generate diverse sets of highly optimized molecules tailored to specific applications [12–14]. The application of generative models in molecular design has

witnessed an exponential growth in recent years, leading to a diverse array of models with varying molecular representations, architectures, and target design problems. To facilitate meaningful comparisons among the expanding number of generative models, several benchmarks have been proposed. These benchmarks evaluate models based on factors such as distribution learning, chemical diversity, and novelty [15, 16]. Additionally, the field of generative models in molecular design has been extensively reviewed, providing comprehensive summaries of its development [4, 17–21].

Despite these remarkable advancements, the practical application of generative models in discovering molecules for real-world scenarios remains limited. Many studies primarily focus on optimizing computational metrics like logP (partition coefficient logarithm) or QED [22] (quantitative estimate of drug-likeness), with only a few published works involving experimental validation of the identified lead molecules. As a result, there are clear gaps that hinder generative models from fully realizing their potential in molecular discovery. Closing these gaps and expanding the scope of applications is crucial for the future advancement of generative models in this domain.

This chapter endeavors to provide a lucid and informative analysis of the recent progress in utilizing generative models for de novo design. Our objective is to present the latest state-of-the-art models and methods, with a focus on their practical applications demonstrated through compelling case studies. Additionally, we will delve into the challenges that currently exist and outline the future directions that hold promise for advancements in this dynamic field.

2 The Generative Model’s Toolkit

2.1 *The Power of Big Data in Drug Discovery*

In today’s data-driven world, the term “big data” has gained significant traction. Yet, it’s remarkable to note that a universally accepted definition of big data remains elusive. Big data refer to exceptionally large and complex datasets, characterized by their diversity, heterogeneity, complexity, and sheer scale. These extensive datasets often exhibit multidimensional properties and can exist in structured or unstructured formats. They are meticulously collected from a wide array of sources, encompassing genomic profiling, pharmacokinetic (PK) and pharmacodynamics (PD) properties of chemical compounds, randomized controlled trials (RCTs), electronic health records (EHRs), medical records, medical claims, product and disease registries, patient-reported outcomes (PROs), and the ever-growing universe of health-monitoring devices.

To fully capitalize on the potential of big data, the adoption of new technological platforms and specialized analytical techniques becomes paramount. Traditional approaches prove inadequate when confronted with the complexities of big data. Therefore, leveraging innovative methodologies such as machine learning algorithms

and advanced analytics is crucial for the curation, control, and analysis of these expansive datasets. These transformative techniques empower researchers and practitioners to derive reliable and actionable insights that were previously unattainable using limited data sets and standard analytical approaches, as highlighted by Berger and Doban [23].

The field of drug discovery faces a paradigm shift driven by the exponential growth in the volume, variety, and accessibility of biological data (Table 1). This deluge of information, encompassing binding affinity, functional activity, ADMET (absorption, distribution, metabolism, excretion, and toxicity), and protein–ligand structures, presents both challenges and unprecedented opportunities to understand the intricate underpinnings of diseases [24–26]. However, harnessing the potential of data science, AI and machine learning becomes instrumental in unlocking the transformative power of these datasets and expediting the drug discovery process [25, 27]. For instance, ML methods offer the ability to predict crucial biological properties of compounds, such as efficacy and toxicity. By analyzing compound features and target characteristics, these methods significantly reduce the cost associated with target identification [26]. Moreover, AI can be employed to predict 3D protein structures, facilitating the identification of potential therapeutics for diseases believed to arise from misfolded proteins [28]. By integrating complex sources of information, such as demographic data, laboratory tests, omics data, imaging data, and physician notes, through AI-driven approaches, novel insights and unexplored hypotheses beyond human capacity can be unearthed [29].

2.2 *Molecular Representations*

Neural networks possess a remarkable ability to handle complex input representations and convert them into latent representations necessary for solving specific tasks. The selection of an appropriate input representation plays a pivotal role in shaping how the model learns and extracts information about the molecule.

Currently, molecular representations can be broadly categorized into three main types:

- (1) One-Dimensional (1-D) Representations. These representations are typically string-based and involve encoding the molecular structure as a linear sequence of characters or tokens. An example of such a representation is SMILES (Simplified Molecular Input Line Entry System) strings [43], which capture the sequential arrangement of atoms and bonds in a molecule (Fig. 1a). SMILES strings have often been utilized in molecular generation tasks and are well-suited for recurrent neural networks [44, 45]. However, challenges related to the validity of SMILES have been observed. Firstly, SMILES must be syntactically valid, ensuring correct pairing of branch parentheses and other grammar rules. For instance, “C(C)” is an invalid SMILES due to incorrect parentheses pairing. Secondly, semantic validity must be maintained, respecting valence constraints

Table 1 Most popular databases for drug discovery

Database	Description	Link	Refs.
ChEMBL	Collection of bioactivity data, including information on compounds efficacy, potency, and selectivity	https://www.ebi.ac.uk/chembl/	[30, 31]
ZINC	Collection of commercially available compounds, providing a valuable resource for virtual screening and lead identification	https://zinc.docking.org	[32, 33]
RCSB PDB	Database providing access to 3D structures of biological macromolecules and their related information	https://www.rcsb.org	[34, 35]
PDBbind	Database containing information about protein–ligand interactions, including binding affinities and structural details of protein–ligand complexes	http://www.pdbbind.org.cn	[36, 37]
Enamine	Collection of commercially available small organic compounds used for virtual screening and drug discovery research	https://enamine.net	[38]
QM9	Public dataset of organic molecules containing up to nine non-hydrogen atoms annotated with quantum chemistry properties	http://quantum-medicine.org/assets/	[39]
PubChem	Public repository of small molecules and their biological properties	https://pubchem.ncbi.nlm.nih.gov	[40]
GDB-13	Database of computationally generated small organic molecules containing up to 13 atoms	https://gdb.unibe.ch/downloads/	[41]
GDB-17	Database of computationally generated small organic molecules containing up to 17 atoms	https://gdb.unibe.ch/downloads/	[41]
DrugBank	Database containing information on drugs and drug targets	https://go.drugbank.com	[42]

and other chemical rules. For example, “O(C)(C)(C)” is syntactically valid but not semantically valid.

Several approaches have been proposed to enhance the validity of SMILES solutions, such as constructing solutions based on a context-free grammar [12, 46, 47] or using attribute grammar [48, 49]. Another approach involves employing machine learning models to predict the invalidity of SMILES prefixes, enabling the discarding of invalid solutions [50]. Although these approaches significantly improve the proportion of valid solutions, they do not provide a guarantee for semantic validity.

To address the validity limitations of SMILES, alternative text representations have been proposed. DeepSMILES, a variant of SMILES, promotes syntactic validity by representing branching and ring closure information in a single location [51].

A more robust approach named SELFIES has also been proposed, ensuring the syntactic and semantic validity of the represented solutions [52]. Cycles and branches

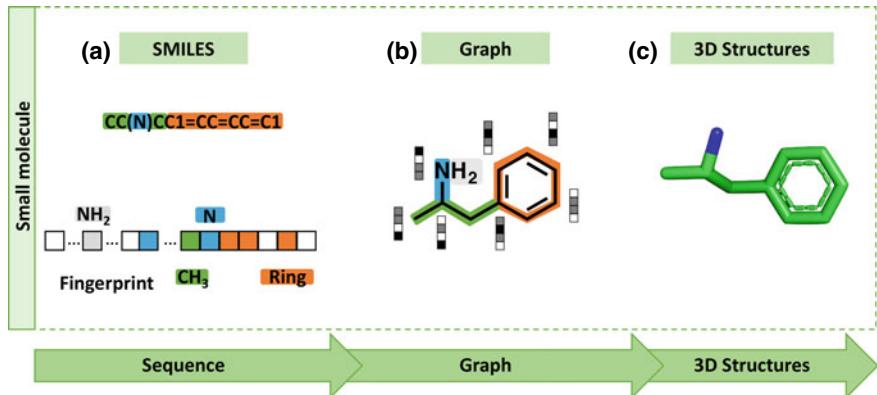


Fig. 1 Schematization illustrating three exemplary molecular representations: **a** one-dimensional (1D) sequence-based representation; **b** graph-based representation; and **c** 3D representation for small molecules

are represented using a start symbol and size symbol. To ensure semantic validity, the decoder interprets and disregards symbols that lead to invalid solutions. For example, a symbol representing the formation of a cycle is ignored if one of the involved atoms lacks the necessary valence electrons. By moving the validity constraints into the decoder, SELFIES provide a more reliable solution. However, it is important to note that certain molecules can be described by an infinite number of SELFIES, such as [N][#N], [N][#N][C], and [N][#N]...[C], all representing dinitrogen, regardless of the number of consecutive atoms at the end of the chain.

Molecular fingerprints represent a widely employed class of descriptors in which structural features are encoded as bits in a bit string or counts in a count vector of fixed length, thus capturing the main structural characteristics and chemical properties [53]. However, their generation process is time-consuming for large databases and is susceptible of possible biases. In the context of generative models, one shortcoming of generating fingerprints as output is that they cannot be directly converted into real molecules. Therefore, fingerprints have been suggested to be better suited for similarity-based screening rather than designing from scratch [19].

- (2) Two-Dimensional (2-D) Representations. In this representation, molecular structures are captured as graphs, with atoms represented as nodes and chemical bonds as edges connecting these nodes (Fig. 1b). An adjacency matrix is commonly used to implement these molecular graphs, specifying atom connectivity and the corresponding bond order/type. Furthermore, nodes and edges can possess associated properties, such as relative spatial location and bond order/type, allowing the encoding of detailed topological data in a processable form [54].

Molecular graphs provide expressive representations that, when coupled with a set of explicit perturbation operators, offer versatile capabilities. They enable the definition of localized perturbations, such as atom substitution [55], as well as perturbations involving chemical functions through the use of fragments (subgraphs) [56] and recombination between molecules [57]. Recombination entails the assembly of fragments from multiple solutions to generate new ones. Deep neural networks can effectively leverage molecular graphs, employing techniques like graph convolutions [58]. However, it is important to note a significant drawback of this representation: the challenge posed by graph isomorphism. Graph isomorphism refers to the task of determining whether two graphs are identical, and it is a known computationally difficult problem. Currently, no efficient algorithm exists to solve graph isomorphism in the general case [59]. As a consequence, verifying the equality of two large molecules represented as graphs can be computationally expensive. Despite this limitation, molecular graphs remain a powerful tool in various molecular modeling tasks. Their ability to encode detailed topological information and facilitate local and global perturbations makes them particularly useful for tasks such as molecular property prediction, reaction prediction, and virtual screening. In the subsequent sections, we will delve deeper into the applications and challenges associated with molecular graph representations, shedding light on their potential benefits and considerations for their utilization in neural network-based approaches.

- (3) Three-Dimensional (3-D) Representations. These representations offer a means to portray molecules within a spatial framework by utilizing point clouds, where each atom is depicted as a point in three-dimensional space (Fig. 1c). They not only capture the covalent atom connectivity but also provide valuable insights into the molecule's conformational preferences. By visualizing the molecule's 3D structure, a deeper understanding and appreciation of its spatial arrangement can be achieved. Prominent examples of such representations include the work by Gebauer et al. [60], who generated molecules sequentially by placing atoms in Cartesian coordinates, and the approach by Simm et al. [61], which adopted a similar strategy by generating molecules atom-by-atom based on internal coordinates. However, there are certain drawbacks associated with using these methods for molecular discovery applications. Firstly, to accurately account for the physical properties of a given molecule, it becomes necessary to consider multiple conformers. This requirement poses a challenge in terms of computational complexity, as obtaining multiple conformers for each molecule can be time-consuming, particularly for large and flexible molecules. To address these limitations, researchers have recently explored alternative approaches. One approach involves replacing or supplementing traditional conformer generation methods [62, 63] with three-dimensional generative models [64–66]. These models are trained on extensive datasets comprising conformer ensembles [39, 67], thereby mitigating the computational expense associated with acquiring three-dimensional training data.

3 Generative Models: A Brief Overview

In this section, we will explore generative models, classifying them into four main categories: recurrent neural network (RNN)-based models, autoencoder (AE)-based models, generative adversarial network (GAN)-based models. Additionally, we will also discuss transfer learning (TL) and reinforcement learning (RL), powerful techniques used to fine-tune the models for generating molecules with desired properties and optimizing compounds like fragments. Before diving into these architectures, we will first touch upon the early pioneers of artificial neural networks, which have paved the way for their development. The fundamental principles and recent advancements of these widely used generative models will be explored in detail.

3.1 Artificial Neural Networks

Artificial neural networks mimic the computational abilities of the human brain and are adept at processing large datasets [68]. A basic neural network comprises three interconnected layers: the input layer, the output layer, and one or more hidden layers. The input layer contains ‘n’ neurons that hold the input information. These neurons are multiplied by random weights, and the resulting values are aggregated in the hidden layer(s). The output of the network is determined by a nonlinear function known as the activation function.

Neural networks receive input signals, perform mathematical transformations, and generate output signals. During the training process, the network adjusts the weights assigned to connections through optimization algorithms like gradient descent, aiming to minimize the disparity between anticipated and desired outputs. Through this iterative process, neural networks uncover intricate patterns and relationships within the data, enhancing their learning capabilities.

To construct a deep learning model, different training algorithms are employed. Deep learning models have the ability to learn autonomously through effective training, necessitating a robust training process and a substantial amount of data. However, it is important to note that no single deep learning model is perfect for all applications. The choice of an appropriate deep learning model depends on the specific requirements and characteristics of the target application.

3.2 (RNN)-Based Models

RNNs are a class of neural networks with recurrent connections that can store and remember complex data for future processing, enhancing prediction capabilities [9, 69]. For an RNN to function properly, it should include at least one feedback connection in the hidden layer, enabling the activation function to operate as a loop. This

recurrent connection enables the network to learn sequences and process temporal information. Nonlinear processing forms the backbone of the RNN architecture, which is made possible by incorporating loops within the hidden layer and combining it with a Multi-Layer Perceptron (MLP) that has memory capabilities [69].

Figure 2a depicts a simplified form of a recurrent network, illustrating the folded and unfolded RNN architecture over time, with three layers: input, hidden layer (with recurrent connection), and output. The stability and performance of the network rely heavily on the hidden layer, particularly when the initialization element is non-zero [70]. Over time, the network consistently learns sequences, with the hidden layer predicting outputs and storing predictions for future timestamps [71]. Nonlinear functions are applied at each timestamp.

If a RNN is well-trained, even with a simple architecture, it can effectively model rich nonlinear dynamics. Since RNNs are suited for nonlinear operations, several parameters are essential for considering them as dynamic systems, including

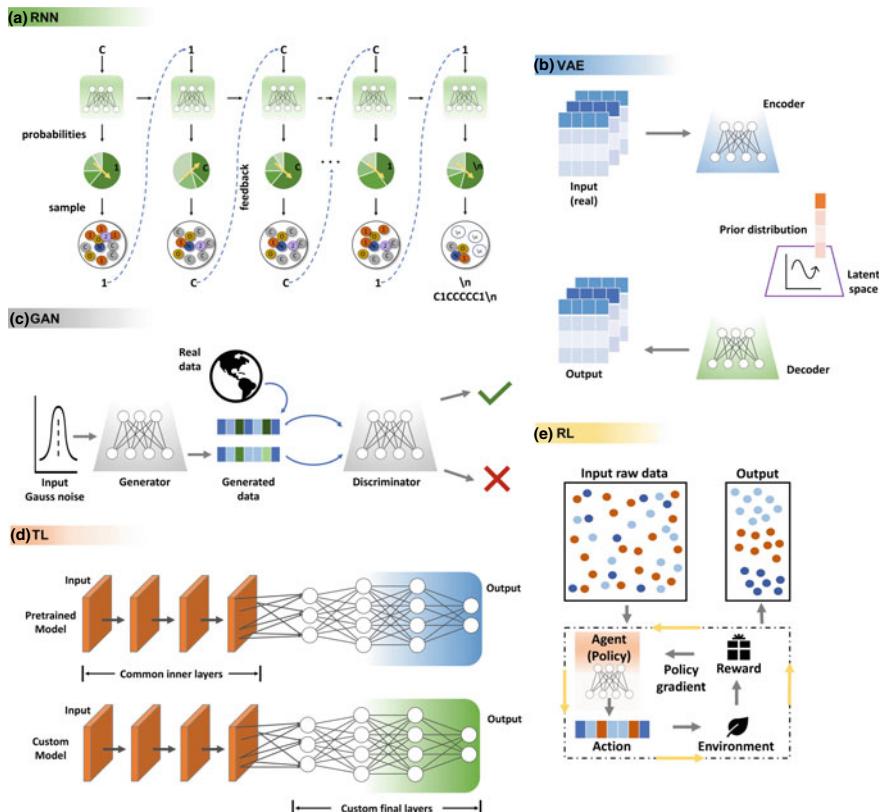


Fig. 2 Schematization of the theory framework of generative models widely employed in drug discovery. **a** RNN, Recurrent Neural Networks; **b** VAE, Variational Autoencoder; **c** GAN, Generative Adversarial Networks; **d** TL, Transfer Learning; **e** RL, Reinforcement Learning

observability, which involves determining the network state based on input or output measurements, and controllability, which ensures that any desired state within a hidden layer can be selected as the initial state throughout time.

In de novo drug design, the application of RNN models allows molecules to be represented as sequences, such as by employing SMILES. By training the RNN model on a substantial collection of SMILES strings, it becomes capable of generating novel and valid SMILES sequences that were not present in the original dataset. This capability positions the RNN model as a generative model for molecular structures. For this purpose, advanced variants of RNNs such as Long Short-Term Memory (LSTM) [72] and Gated Recurrent Unit (GRU) [73] have been introduced. These variants incorporate gate mechanisms to retain crucial input information over extended time steps, addressing a limitation of traditional RNNs. The preference between LSTM and GRU may vary depending on the specific application. While LSTM cells can preserve longer historical context, the additional parameters in LSTM may increase the risk of overfitting. RNNs employing LSTM or GRU have shown great promise in generating de novo small molecules using the SMILES representation [16].

3.3 (AE)-Based Models

AEs are neural networks trained to transform input into output while avoiding simple identity function learning. They consist of an encoder that transforms the input into a compact latent state and a decoder that reconstructs the input from this representation. Training both parts minimizes information loss during reconstruction [6, 74].

As shown in Fig. 2b, variational AEs (VAEs) are a specialized type of AE that assumes data is sampled from a statistical distribution. The encoder transforms the input, such as a SMILES string, molecular fingerprint, or set of molecule features, into parameters of a multidimensional statistical distribution (e.g., means and standard deviations). Sampling from this distribution produces a point that is then fed into the decoder for reconstruction. The training objective includes terms for penalizing reconstruction errors and constraining the encoded parameters to be close to a normal distribution. The training objective includes terms for reconstruction errors and constraining the encoded parameters to resemble a normal distribution. This stochastic process regularizes the network and helps form a meaningful latent space [74, 75].

VAEs are capable of generating new molecules with desired properties by sampling from the learned latent space distribution. These generated molecules can be assessed using various metrics, such as drug-likeness and similarity to existing active compounds (refer to the metrics section). VAEs excel at identifying intricate correlations in molecular data. By utilizing sampling techniques and representing the latent space as a probability distribution, VAEs enable the synthesis of diverse molecules. This probabilistic framework facilitates targeted exploration of chemical space, potentially leading to the discovery of compounds with specific features.

Adversarial AEs (AAEs) offer an alternative to VAEs by incorporating adversarial training to structure the latent space. The encoder transforms the input into a single point within the latent space, while a discriminator network distinguishes between samples from a prior statistical distribution and encoded points. The encoder can be viewed as a generator engaged in a competition with the discriminator, striking a balance between reconstruction and adversarial error [76].

3.4 GAN-Based Models

The invention of GANs in 2014 by Goodfellow sparked a wave of generative models [77]. Inspired by the game theory concept of two-person zero-sum game [77], GANs operate differently from VAEs. They don't rely on an explicit probability density function (Fig. 2c), but instead utilize a generator-discriminator framework. The discriminator is trained to classify synthetic molecules generated by the generator, aiming to maximize the error rate and differentiate them from real data. Through an adversarial, zero-sum game, the generator and discriminator are jointly trained until the discriminator is deceived, indicating that the generator produces plausible molecules.

A variant of GANs, called Conditional GANs (CGAN) [78], introduces a change in the input requirements. While the original GAN only requires input data, CGAN incorporates additional conditions, such as label information, during training. This label information can represent categories or parameters associated with real data. After training, the CGAN model can generate synthetic data based on specific conditions and random noise. This capability is particularly valuable when users want to generate synthetic data belonging to a particular class or adhere to specific physical parameters.

3.5 Transfer Learning

Deep supervised neural network training typically requires a substantial amount of labeled data. However, when labeling data at such a large scale is not feasible, TL provides a promising solution [79]. The essence of TL lies in adopting a pretrained model trained on a vast dataset and fine-tuning it for a specific task using a smaller available dataset (Fig. 2d). Leveraging the knowledge acquired during pretraining significantly reduces the training time required on the final dataset and often leads to improved performance. Furthermore, active learning, as demonstrated by Houlsby et al. [80], facilitates a drastic reduction in the amount of labeled data by carefully selecting instances for annotation.

TL has found valuable applications in chemoinformatics, particularly in the domain of improved molecular representations. The representation of molecules

in a machine-readable format has been a long-standing area of research in chemoinformatics. Traditionally, molecules were represented using sparse vectors based on various types of fingerprints [81] and molecular descriptors [82]. These representations encoded different features, such as the presence of functional groups or molecular weight, with each element in the vector holding a specific value.

Inspired by recent advances in natural language processing (NLP) that have successfully utilized TL, researchers have explored the application of neural network-based representations to molecular compounds [83]. Through this approach, dense vector representations, known as molecular embeddings, are learned by extracting latent representations from a large repository of compounds, even if they are not directly related to the specific task of interest. When a trained neural network is presented with a new compound, it can generate the corresponding representation.

Similar to other TL applications, the effectiveness of molecular embeddings rests on the hypothesis that the latent features learned from self-supervised tasks, such as predicting the next token or a masked token, capture relevant structural information that can be useful for various tasks. ChemBERTa, an exemplary molecular embedding model, is based on a BERT-like transformer architecture [84]. It demonstrates the potential of leveraging TL techniques to enhance the representation of molecular compounds.

3.6 Reinforcement Learning

In recent years, deep RL has emerged as a powerful framework for optimizing objectives, showcasing remarkable achievements like AlphaGo [85]. The vastness of the chemical space, akin to the immense solution space of the game Go, makes RL a promising approach for navigating and exploring chemical landscapes through dynamic decision-making processes [86].

As illustrated in Fig. 2e, RL involves three key components: an agent, a reward function, and an environment. The goal is to optimize the agent's behavior towards a user-defined target. The agent selects the next action, while the reward function evaluates the quality of these actions based on domain-specific rules and provides feedback to the agent. To leverage RL effectively, a generative model is initially trained on a comprehensive dataset of molecules to learn the grammar of SMILES representations. Once trained, RL serves as a valuable technique for fine-tuning specific properties, such as synthetic accessibility [87] and quantitative estimates of druglikeness [88], which evaluate various physical attributes.

For instance, a method called policy gradient for forward synthesis (PGFS) utilizes RL to generate synthetically accessible molecules [89]. In this method, the agent is represented by a neural network, and the policy actions involve chemical transformations, such as adding or removing atoms and bonds. The reward signal used in PGFS is based on the measure of synthetic accessibility [89]. By integrating RL with generative models, researchers gain a powerful methodology for exploring and

optimizing the chemical space, allowing them to generate molecules with desired properties and enhance their chemical design endeavors.

3.7 Evaluation Metrics

In order to assess the quality of molecules designed by generative models, and to gauge the generative model performances, a variety of evaluation methods have been introduced as benchmarking metrics, also embedded into platforms (Table 2) [15, 16].

Molecule validity indicates that the generated molecules follow the rules of chemistry, such as valid atom and bond types and valence criteria. Validity ensures that the produced structures are thus chemically feasible. Further and stricter rule-based filters (such as the maximum ring size filter by Walter, https://github.com/PatWalters/rd_filters) can also be implemented to shortlist the valid molecules.

Among the valid molecules, “uniqueness” evaluates the percentage of non-redundant ones, whereas “novelty” evaluates the fraction of generated molecules that are not present in the training set or other reference datasets, thus assessing the ability of the model to generate new chemical matter. In this regard, molecular diversity can be also evaluated on the basis of fragment [90] or scaffold similarity [91], or using Frechet ChemNet Distance (FCD) [92], which evaluates the properties (chemical or biological) distribution of generated molecules and real ones.

Depending on the specific research project goals, further metrics might be useful, in order to decide the most promising candidates to be advanced in the next phases of drug development.

The “quantitative estimation of drug-likeness” (QED) is a metric for assessing drug-likeness of generated molecules [22]. It is based on “desirability functions” which were fitted to the distributions of the molecular properties (mostly physico-chemical descriptors) of marketed oral drugs. The higher this score, the higher is the attractiveness of compounds as “hits”, on which to undertake therapeutic development. In a similar fashion, the “quantitative estimate of protein–protein interactions” (QEPII) provides an estimation of the compound’s ability to target protein–protein interactions [93]. The “synthetic accessibility score” (SAS) determines how the generated compounds are difficult to synthesize [87]. A SAS value higher than 5 denotes a compound that is more challenging to synthesize. The MOSES [15] and GuacaMol [16] platforms have been conceptualized to allow, from one side, the evaluation of generated molecules on the basis of the most widely used above-described metrics, and also to enable comparisons of a generative model against other “baseline” models, to better validate its performances. Finally, property predictions (for instance, predicted bioactivity against a given molecular target) can be regarded as a further metric to be satisfied by the generated compounds [17]. In this regard, quantitative structure–activity relationship (QSAR) models or more “classical” ML models are often employed at the final stage to assess in silico compound properties.

Table 2 A list of evaluation metrics for molecular generative models

Metrics	Description	Refs.
Validity	Assesses the consistency of atom and bond types and valence criteria	[16]
Uniqueness	Evaluates the uniqueness of created compounds by comparing them to existing compounds in reference datasets	[16]
Novelty	Measures how much novel structural information or chemical patterns are introduced by the created molecules compared to the reference dataset	[16]
Quantitative estimation of drug-likeness (QED)	Calculates the probability that a chemical will act as a drug by combining several molecular characteristics, including molecular weight, lipophilicity, and synthetic accessibility	[22]
Synthetic accessibility score (SAS)	Determines how simple it is to synthesize a molecule. It considers elements like structural complexity, accessibility of raw resources, and necessary synthetic procedures	[87]
Quantitative estimate of protein-protein interactions (QEPII)	Predicts the potential ability of a compound to target protein-protein interactions	[93]
Fréchet ChemNet distance (FCD)	Metric used to measure the similarity between two distributions of chemical compounds through determining the minimum separation between the distributions	[92]

4 Applications of Generative Models in Drug Design

As detailed above, generative models are able to learn chemical patterns from existing molecules (usually retrieved from chemical databases) and use these patterns to construct new chemical entities. One of the first contributions in the field came from Gomez-Bombarelli et al. [44], which combined a VAE model (consisting of an encoder, translating SMILES strings into continuous vectors in latent space, and a decoder converting such vectors to SMILES again) and a molecular properties predictor. This model, however, in many cases yielded invalid chemical structures. Another model that has been widely used for molecule generation is RNN [19], which can sample from the distributions learned from the input molecules to produce new ones in the form of SMILES. In the REINVENT model, which is also based on RNN, a RL method was implemented to fine-tune the pre-trained RNN, so the model could generate structures with desirable properties [45, 94]. In the following sections, we focus on the most recent practical applications of generative models, with particular emphasis on case studies either addressing underexplored areas of de novo drug design or performing extensive experimental validation.

4.1 3D Generative Models

The majority of generative models proposed so far have been mostly “ligand-centric”, i.e., they have been built on 2D ligand information (also extensively reviewed in [17, 19]), without taking into account any information on the ligands’ binding sites and thus loosing possibly useful information of protein binding pockets. One of the first attempts is represented by the work of Skalic et al. [95], who developed a GAN-based model to generate diverse three-dimensional ligand shapes complementary to a protein pocket. Ragoza et al. [96] developed a model based on a conditional VAE, trained on a grid of atomic densities of cross-docked protein–ligand structures. The generated atomic densities were subsequently employed to build valid molecular conformations following atom-fitting iterations and bond-inference procedures. In the RELATION model [97], also based on VAE, the geometric features of the protein–ligand complexes are transferred to a latent space for generation. Then, the generation of molecules with optimal conformations and pharmacophoric features is informed by pharmacophore conditioning and docking-based Bayesian sampling. RELATION was subsequently implemented in a webservice called ReMODE (Receptor-based MOlecular DEsign) [98], in which the prospective user could also control physicochemical properties and optimize parameters such as drug-likeness or synthetic accessibility of the generated molecules. The STRIFE (Structure Informed Fragment Elaboration) approach [99] by using a constrained graph VAE, takes as input a fragment hotspot maps (FHM)s from a protein target of interest to quickly construct fragment elaborations with complementary pharmacophores to the protein. For a more comprehensive overview of 3D molecular generative models, the reader is referred to a recent review [100]. Very recently, Zhang et al. [101] suggested a conditional RNN (cRNN)-based generative model coupled with a ligand–protein interaction finger-print (IFP) as constraints to produce new molecules with desired binding affinity to the active site (Fig. 3). The IFPs were constructed on ligand docking poses into the protein pocket, encoding five types of interactions (H-bond, halogen bond, aromatic, electrostatic, and hydrophobic). The authors selected two proteins as case study, namely CDK2 (Cyclin-dependent kinase 2) and A2A (Adenosine A2A Receptor). As training set, they used 300,000 compounds randomly selected from ChEMBL and a target active set made up by 20 compounds active on CDK2 used as “seeds”. The cRNN allows to integrate molecular properties as constraints; in this way, the generated compounds are forced to follow the constraints of the input. In this case, the IFPs constraints are taken as input, and each IFP corresponds to a sub-chemical space consisting of compounds that have similar IFP. Thus, sampling the particular chemical space is realized by sampling the IFP. The compounds generated with this model yielded better docking scores than compounds sampled using REINVENT as baseline model [94] and a random set of compounds from ChEMBL, used as benchmark; however, the compounds showed a slightly higher docking score than 611 CDK2 active compounds with IC_{50} lower than 50 nM from ChEMBL. The authors then explored different options in terms of number of seed compounds (i.e. using only one compound co-crystallized with the protein of interest) and additional properties

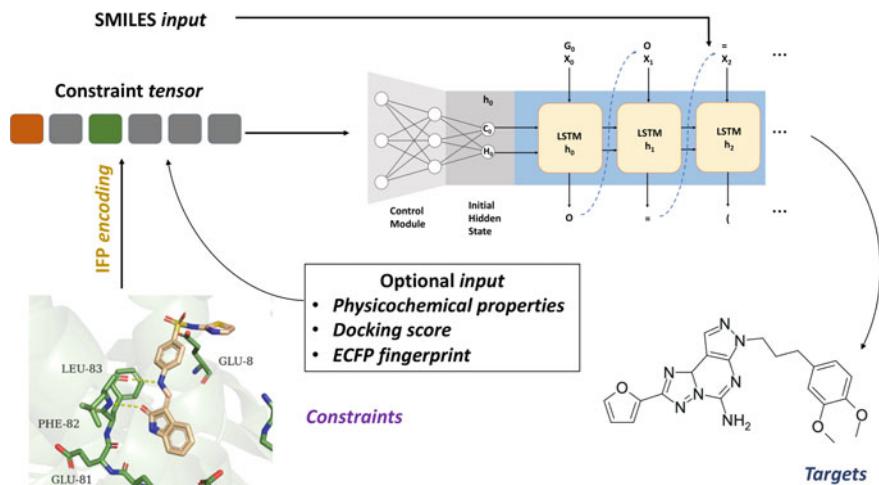


Fig. 3 Workflow of the cRNN model. The protein-ligand interactions were encoded as an IFP vector. The IFPs were used as constraints for compounds generation

to be used as constraints in composite models (including the physico-chemical properties; ECFP fingerprints; SMILES), as well as various metrics to judge the overall performances, namely, docking score, IFP recovery, IFP similarity, and molecular similarity to the seed. The composite model ECFP + AIFP performed well in almost all metrics, also generating compounds more similar to the seed. However, in the context of lead identification, higher novelty and diversity may be more desirable. Models based on IFP, instead, displayed lower molecular similarity to the seed. The method is highly dependent on the accuracy of the docking poses, thus, if the poses are inaccurate and the scores are unfavorable, the molecules generated could be invalid.

An ever-increasing number of studies started to adopt molecular docking as a filter to further evaluate de novo generated compounds, relying on the docking scores to estimate the potential binding affinity to the target of interest, or to guide the generative process. For these reasons, docking has been even claimed as a possible benchmark to be used in combination with the above-described metrics [102]. In addition, docking engines are being included in the framework of generative models: for example, DockStream, embedded into the REINVENT ecosystem [103], is a docking wrapper providing access to a collection of docking backends; DOCKSTRING is a package for computation of docking scores in the context of models' comparisons [104]. There is certainly room for further improvements, however, we can anticipate that molecular generation “conditioned” on receptor structures has great potential, enabling to directly design drug-like molecules in a target conditioning manner.

4.2 Success Stories in Early Drug Discovery

Generative models routinely undergo retrospective validation, usually by employing test sets made up of a fraction of training data reserved for this purpose. Prospective validation via experimental testing has been more rarely addressed, however it could also provide opportunities to improve the model. While earlier studies were mostly “theoretical”, we are now starting to see more prospective applications of generative models in the context of drug discovery campaigns, with the generated compounds being evaluated both *in vitro* and *in vivo*.

Zan et al. [105] integrated a deep generative scaffold decorator model [106] with molecular docking for designing novel discoidin domain receptor 1 (DDR1) inhibitors (Fig. 4). They collected a training set comprised of 3603 million molecular pairs obtained from slicing a set of 902 DDR/FGFR (fibroblast growth factor receptor) inhibitors, retrieved from ChEMBL, by using the Matched Molecular Pairs (MMP) algorithm. The slicing procedure allowed to obtain a vocabulary of possible substitutions (qualifying as scaffold decorations tuples). To better distinguish scaffolds from decorations, they were required to possess at least one ring and to meet specific physicochemical properties. Then, SMILES string’s attachment points were consecutively decorated using an encoder-decoder generative model. Initially, the model decorated the first attachment point, and subsequently, the partially constructed molecule was reintroduced as a new scaffold for further decoration. This iterative process progressed until all attachment points were decorated. Variability in the order of attachment sites was also introduced by randomizing the SMILES string at each stage, so that the generative process increases the diversity and coverage of the generated compounds. The generated library was then triaged using several filters, including removal of “pan-assay interference compounds” and other structural alerts, selected physico-chemical properties, kinase selectivity screening as well as docking. Finally, two compounds were selected for synthesis and bioactivity experiments. Compound **1** and compound **2**, showed potent inhibitory activity against DDR1 with IC₅₀ values of 10.2 ± 1.2 nM and 10.6 ± 1.9 nM, respectively. Furthermore, compound **2** displayed exceptional kinase selectivity, displayed reasonable synthetic accessibility scores and clogP values. One possible disadvantage is that the produced compounds frequently reached higher molecular weights than known DDR1 inhibitors. Potential cross-reactivity was another limitation, because some of the generated compounds displayed predicted activity for other kinases. Remarkably, the generated compound **2** inhibited the expression of pro-inflammatory cytokines and DDR1 autophosphorylation in cells. Since DDR1 is expressed in epithelial cells of a variety of tissues, including intestinal tissues, **2** was tested in an inflammatory bowel disease (IBD) mouse model, thereby showing significant therapeutic protection.

Li et al. recently proposed a cRNN model to build a virtual library of target-specific compounds [107]. RIPK1, a protein kinase involved in cell survival and associated to inflammatory and autoimmune diseases, was selected as a promising target for the application of this model. In order to enhance the ability of the model to enrich

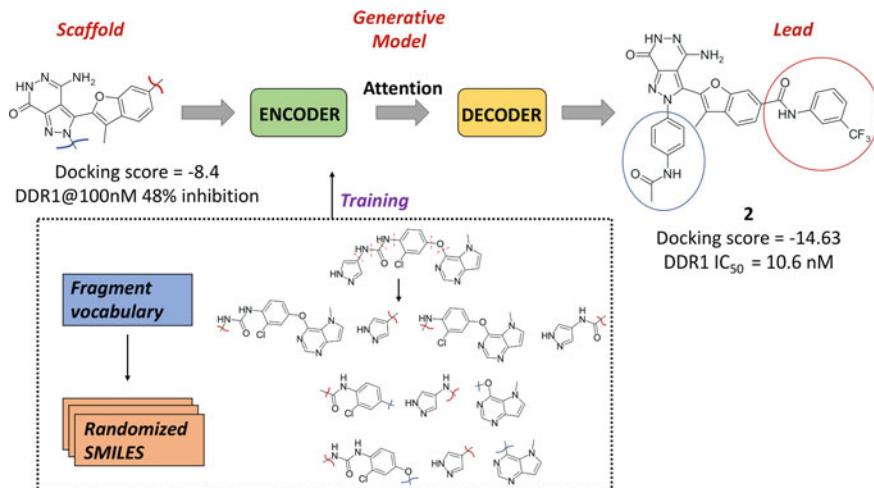


Fig. 4 Workflow of the scaffold-decorating model. The generated selective DDR1 lead candidate **2** is shown on the right side

the virtual library of novel molecular scaffolds, the authors adopted three strategies including TL, regularization enhancement, and sampling enhancement. TL was employed to shift the data distribution of the latent space from the large collection of the source data (~16 million molecules from ZINC12 database) toward the target data (1030 known RIPK1 inhibitors with an activity threshold of 10 mM retrieved from ChEMBL and patent literature). Regularization enhancement by adding a Gaussian noise vector, was used to improve the generalization performance of the model, being considered similar to introducing penalty terms in the objective function. Sampling enhancement was implemented by interpolating between latent space during model generation, to augment the chances of generating target-specific molecules bearing as much diverse chemical scaffolds as possible. The generative model produced 79,323 molecules that were specifically designed as a virtual compound library for RIPK1. When compared to the source and target data, the generated compounds showed remarkable diversity, measured in terms of new scaffolds. Virtual screening based on pharmacophore screening and subsequent docking allowed to prioritize eight compounds for chemical synthesis and bioactivity evaluation. Among these eight compounds, RI-962 exhibited the most potent inhibitory activity against RIPK1 with an IC₅₀ value of 35 nM, and high selectivity against a panel of different kinases. Interestingly, the X-ray crystal structure of RIPK1 in complex with RI-962 was also obtained at a resolution of 2.64 Å. RI-962 showed good in vivo efficacy in two inflammatory models, namely the TNFα-induced systemic inflammatory response syndrome (SIRS) and dextran sulfate sodium (DSS)-induced inflammatory bowel disease (IBD).

De novo design based on generative models have mostly focused on target proteins with many known ligands. Therefore, underexplored targets, such as orphan receptors, represent a challenging scenario, given the scarcity of active ligands knowledge. This challenge has been recently addressed by Ballarotto et al., who focused on the orphan nuclear receptor related 1 (Nurr1), a neuroprotective transcription factor showing promises for the treatment of conditions like Alzheimer's, Parkinson's, and multiple sclerosis [108]. However, there are very few strong Nurr1 activators. The same authors previously identified an agonist, compound **1** ($EC_{50} = 0.4 \mu M$, $K_d = 0.7 \mu M$) which they used as "template" together with further five weaker but structurally similar ligands, in a data augmentation effort. The authors used a previously reported model, defined as CLM (chemical language model) [109, 110]. Basically, CLM implements a RNN with long short-term memory (LSTM); these models proved to be particularly successful in proposing new chemical entities with potential bioactivity (Fig. 5). This has been achieved via TL,11,12 which uses a model previously trained on a large set of data and "fine-tunes" it using a target-focused (and smaller) set of molecules. The CLM was previously trained using 365,000 molecules from ChEMBL, in order to capture the syntax of SMILES strings and generic molecular characteristics. Then, a two-stage fine tuning procedure was followed; a beam search, a heuristic sampling technique, was employed to identify training epochs where the model was predisposed to produce molecules similar to compound **1**. The choice of the compounds for synthesis was based on two selection criteria. The first one was the sampling frequency of each molecule, which provided the CLM with an internal metric of priority. The model assumed that molecules with greater sampling frequencies had more likely configurations. According to the second criterion, which was based on physicochemical descriptors and an examination of molecular shape, molecules that showed a high degree of resemblance to compound **1** were chosen. Interestingly, the highly frequent sampled molecules (compounds **7–9**) were characterized by lower similarity to **1**, while the molecules **10–12** showing a higher degree of similarity to **1**, were not frequently sampled. Six de novo designed compounds were finally synthesized and tested to evaluate Nurr1 agonism, and two of them displayed the desired bioactivity (compound **7** with $EC_{50} = 0.03 \mu M$ and compound **8** with $EC_{50} = 2.1 \mu M$). Compound **7** in particular showed high structural novelty compared to the known Nurr1 ligands, and also exhibited remarkable agonist potency, becoming one of the most potent Nurr1 agonists discovered thus far.

5 Concluding Remarks and Future Perspectives

In this chapter, the most relevant generative models applied to de novo drug design have been portrayed, emphasizing the latest scientific developments and exemplary applications. The role of these approaches is growing, because they are particularly well suited to navigate extremely large chemical spaces and to optimize molecules towards a clinical candidate, more efficiently than human experts and classical virtual

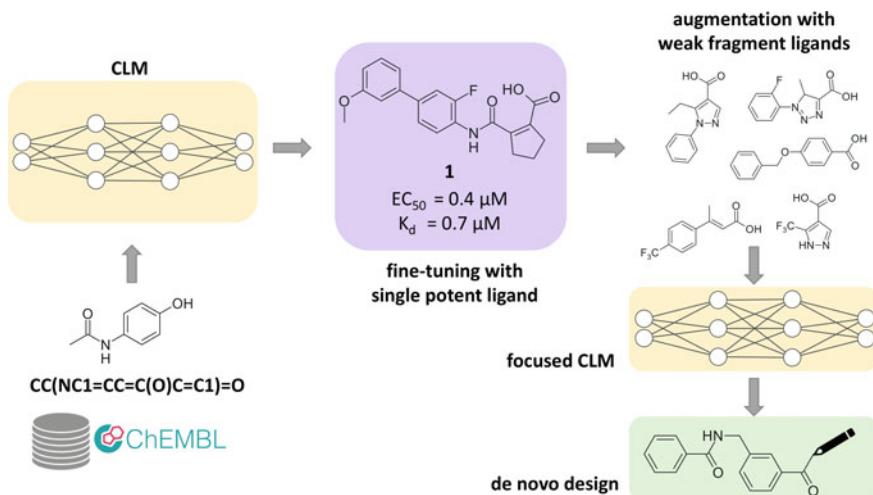


Fig. 5 Workflow of the CLM model for the design of novel Nurr1 agonists

screening. This is also reflected by the higher number of success stories in scientific literature, highlighting how DL methodologies can be employed end-to-end in drug discovery. Still, several challenges remain to be addressed. Regarding generative model validation, we argue that a collective effort should be made to improve benchmarking methods and metrics towards more standardized procedures. Many studies lack a thorough validation of the proposed model, for instance with other baseline models, therefore it is sometimes very difficult to judge if it is truly more advantageous and what its added value is with respect to the state of the art. The currently used metrics to measure the overall model performance also have some shortcomings. In fact, Renz et al. [111], by using a model that simply adds a carbon atom at some random spot in SMILES string, demonstrated that validity, novelty and uniqueness metrics struggle to distinguish generative models from simple text manipulation. Thus, as they stand, these metrics may only offer a cursory understanding of how beneficial such models are for real tasks. In order to improve the reliability of generative models' results, an open-source benchmark for practical molecular optimization (PMO) has been proposed recently [112], as a standardized experimental setup to comprehensively evaluate and compare new molecule optimization methods with existing ones. The drug-likeness and synthesizability scores also have some limitations [111], as highly scoring molecules are often synthetically infeasible or contain unstable functionalities. Therefore, the human expertise still remains crucial when it comes to select the most promising candidates to advance in the next phases of drug development. In order to further support decision-making, AI-based tools to assess synthetic accessibility of the idea generated, and to assist with synthesis planning, are becoming increasingly popular and shall start to be integrated into the generative model pipelines. Looking ahead, we can expect further exciting progresses in this field; for instance, it has been envisaged, as a long-term

goal, the integration of generative models with automated platforms able to synthesize the proposed molecules and carrying out their experimental evaluation [113]. For progress, it will be essential that AI is not perceived as a threat, because for the time being, the knowledge and skills of medicinal chemists remain still crucial; reducing the barriers between different disciplines, such as computer science and chemistry will be of pivotal importance, as well as familiarizing with each other's domains and fostering a new collaborative mindset.

References

1. Kiriiri, G. K., Njogu, P. M., & Mwangi, A. N. (2020). Exploring different approaches to improve the success of drug discovery and development projects: A review. *Futur J Pharm Sci*, 6, 1–12. <https://doi.org/10.1186/s43094-020-00047-9>
2. Waring, M. J., Arrowsmith, J., Leach, A. R., Leeson, P. D., Mandrell, S., Owen, R. M., Pairaudeau, G., Pennie, W. D., Pickett, S. D., Wang, J., Wallace, O., Weir, A. (2015). An analysis of the attrition of drug candidates from four major pharmaceutical companies. *Nature Reviews Drug Discovery*, 14, 475–486. <https://doi.org/10.1038/nrd4609>
3. Lavecchia, A., & Giovanni, C. (2013). Virtual screening strategies in drug discovery: A critical review. *Current Medicinal Chemistry*, 20, 2839–2860.
4. Cerchia, C., & Lavecchia, A. (2023). New avenues in artificial-intelligence-assisted drug discovery. *Drug Discovery Today*, 28, 103516. <https://doi.org/10.1016/j.drudis.2023.103516>
5. Bohacek, R. S., McMurtin, C., & Guida, W. C. (1996). The art and practice of structure-based drug design: A molecular modeling perspective. *Medicinal Research Reviews*, 16, 3–50.
6. Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521, 436–444. <https://doi.org/10.1038/nature14539>
7. Bond-Taylor, S., Leach, A., Long, Y., & Willcocks, C. G. (2022). Deep generative modelling: A comparative review of VAEs, GANs, normalizing flows, energy-based and autoregressive models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44, 7327–7347. <https://doi.org/10.1109/TPAMI.2021.3116668>
8. Radford, A., Metz, L., & Chintala, S. (2016). *Unsupervised representation learning with deep convolutional generative adversarial networks*. 4th International Conference on Learning Representation ICLR 2016—Conf Track Proc.
9. Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., & Bengio, S. (2016). *Generating sentences from a continuous space*. CoNLL 2016—20th SIGNLL Conference on Computational Natural Language Learning Proc, pp. 10–21. <https://doi.org/10.18653/v1/k16-1002>
10. van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., & Kavukcuoglu, K. (2016). *WaveNet: A generative model for raw audio*. arXiv Prepr arXiv160903499
11. Engel, J., Resnick, C., Roberts, A., Dieleman, S., Eck, D., Simonyan, K., & Norouzi, M. (2017). *Neural audio synthesis of musical notes with WaveNet autoencoders*. 34th International Conference on Machine Learning, ICML 2017. PMLR, pp. 1771–1780.
12. Kusner, M. J., Paige, B., & Hemández-Lobato, J. M. (2017). *Grammar variational autoencoder*. 34th International Conference on Machine Learning, ICML 2017. PMLR, pp. 3072–3084.
13. Segler, M. H. S., Kogej, T., Tyrchan, C., & Waller, M. P. (2018). Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Science*, 4, 120–131. <https://doi.org/10.1021/acscentsci.7b00512>
14. Jin, W., Barzilay, R., & Jaakkola, T. (2021). *Chapter 11: Junction tree variational autoencoder for molecular graph generation*. RSC Drug Discovery Series. PMLR, pp. 228–249.

15. Polykovskiy, D., Zhebrak, A., Sanchez-Lengeling, B., Golovanov, S., Tatanov, O., Belyaev, S., Kurbanov, R., Artamonov, A., Aladinskiy, V., Veselov, M., Kadurin, A., Johansson, S., Chen, H., Nikolenko, S., Aspuru-Guzik, A., & Zhavoronkov, A. (2020) Molecular sets (MOSES): A benchmarking platform for molecular generation models. *Frontiers in Pharmacology*, 11.
16. Brown, N., Fiscato, M., Segler, M. H. S., & Vaucher, A. C. (2019). GuacaMol: Benchmarking models for de Novo molecular design. *Journal of Chemical Information and Modeling*, 59, 1096–1108. <https://doi.org/10.1021/acs.jcim.8b00839>
17. Bilodeau, C., Jin, W., Jaakkola, T., Barzilay, R., Jensen, K. F. (2022). Generative models for molecular discovery: Recent advances and challenges. *Wiley Interdisciplinary Reviews Computational Molecular Science*, 12, e1608. <https://doi.org/10.1002/wcms.1608>
18. Cheng, Y., Gong, Y., Liu, Y., Song, B., & Zou, Q. (2021). Molecular design in drug discovery: A comprehensive review of deep generative models. *Briefings in Bioinformatics*, 22, bbab344. <https://doi.org/10.1093/bib/bbab344>
19. Tong, X., Liu, X., Tan, X., Li, X., Jiang, J., Xiong, Z., Xu, T., Jiang, H., Qiao, N., Zheng, M. (2021). Generative models for de Novo drug design. *Journal of Medicinal Chemistry*, 64, 14011–14027. <https://doi.org/10.1021/acs.jmedchem.1c00927>
20. Wang, M., Wang, Z., Sun, H., Wang, J., Shen, C., Weng, G., Chai, X., Li, H., Cao, D., Hou, T. (2022). Deep learning approaches for de novo drug design: An overview. *Current Opinion in Structural Biology*, 72, 135–144. <https://doi.org/10.1016/j.sbi.2021.10.001>
21. Meyers, J., Fabian, B., & Brown, N. (2021). De novo molecular design and generative models. *Drug Discovery Today*, 26, 2707–2715. <https://doi.org/10.1016/j.drudis.2021.05.019>
22. Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S., Hopkins, A. L. (2012). Quantifying the chemical beauty of drugs. *Nature Chemistry*, 4, 90–98. <https://doi.org/10.1038/nchem.1243>
23. Berger, M. L., & Doban, V. (2014). Big data, advanced analytics and the future of comparative effectiveness research. *Journal of Comparative Effectiveness Research*, 3, 167–176. <https://doi.org/10.2217/cer.14.2>
24. Chen, H., Engkvist, O., Wang, Y., Olivecrona, M., Blaschke, T. (2018). The rise of deep learning in drug discovery. *Drug Discovery Today*, 23, 1241–1250. <https://doi.org/10.1016/j.drudis.2018.01.039>
25. Lavecchia, A. (2019). Deep learning in drug discovery: Opportunities, challenges and future prospects. *Drug Discovery Today*, 24, 2017–2032. <https://doi.org/10.1016/j.drudis.2019.07.006>
26. Paul, D., Sanap, G., Shenoy, S., Kalyane, D., Kalia, K., Tekade, R. K. (2021). Artificial intelligence in drug discovery and development. *Drug Discovery Today*, 26, 80–93. <https://doi.org/10.1016/j.drudis.2020.10.010>
27. Lavecchia, A. (2015). Machine-learning approaches in drug discovery: Methods and applications. *Drug Discovery Today*, 20, 318–331. <https://doi.org/10.1016/j.drudis.2014.10.012>
28. Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T. F. G., Qin, C., Zidek, A., Nelson, A., Bridgland, A., Penedones, H., Petersen, S., Simonyan, K., Crossan, S., Jones, D. T., Silver, D., Kavukcuoglu, K., Hassabis, D., Senior, A. W. (2018). De novo structure prediction with deep-learning based scoring. *Thirteen Critical Assessment of Technical Protein Structure*, 2, 1–2.
29. Vamathevan, J., Clark, D., Czodrowski, P., Dunham, I., Ferran, E., Lee, G., Li, B., Madabhushi, A., Shah, P., Spitzer, M., Zhao, S. (2019). Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, 18, 463–477. <https://doi.org/10.1038/s41573-019-0024-5>
30. Gaulton, A., Hersey, A., Nowotka, M., Bento, A. P., Chambers, J., Mendez, D., Mutowo, P., Atkinson, F., Bellis, L. J., Cibrián-Uhalte, E., Davies, M., Dedman, N., Karlsson, A., Magariños, M. P., Overington, J. P., Papadatos, G., Smit, I., Leach, A. R. (2017). The ChEMBL database in 2017. *Nucleic Acids Research*, 45, D945–D954. <https://doi.org/10.1093/nar/gkw1074>
31. Mendez, D., Gaulton, A., Bento, A. P., Chambers, J., De Veij, M., Félix, E., Magariños, M. P., Mosquera, J. F., Mutowo, P., Nowotka, M., Gordillo-Marañón, M., Hunter, F., Junco, L.,

- Mugumbate, G., Rodriguez-Lopez, M., Atkinson, F., Bosc, N., Radoux, C. J., Segura-Cabrera, A., Hersey, A., Leach, A. R. (2019). ChEMBL: Towards direct deposition of bioassay data. *Nucleic Acids Research*, 47, D930–D940. <https://doi.org/10.1093/nar/gky1075>
32. Sterling, T., & Irwin, J. J. (2015). ZINC 15—ligand discovery for everyone. *Journal of Chemical Information and Modeling*, 55, 2324–2337.
33. Irwin, J. J., Tang, K. G., Young, J., Dandarchuluun, C., Wong, B. R., Khurelbaatar, M., Moroz, Y. S., Mayfield, J., Sayle, R. A. (2020). ZINC20—A free ultralarge-scale chemical database for ligand discovery. *Journal of Chemical Information and Modeling*, 60, 6065–6073.
34. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., Bourne, P. E. (2000). The protein data bank. *Nucleic Acids Research*, 28, 235–242. <https://doi.org/10.1093/nar/28.1.235>
35. Berman, H. M. (2008). The protein data bank: A historical perspective. *Acta Crystallographica Section A*, 64, 88–95.
36. Wang, R., Fang, X., Lu, Y., & Wang, S. (2004). The PDBbind database: Collection of binding affinities for protein-ligand complexes with known three-dimensional structures. *Journal of Medicinal Chemistry*, 47, 2977–2980. <https://doi.org/10.1021/jm0305801>
37. Liu, Z., Li, Y., Han, L., Li, J., Liu, J., Zhao, Z., Nie, W., Liu, Y., Wang, R. (2015). PDB-wide collection of binding data: Current status of the PDBbind database. *Bioinformatics*, 31, 405–412.
38. Grygorenko, O. O., Radchenko, D.S., Dziuba, I., Chuprina, A., Gubina, K. E., Moroz, Y. S. (2020). Generating multibillion chemical space of readily accessible screening compounds. *iScience*, 23, 101681. <https://doi.org/10.1016/j.isci.2020.101681>
39. Ramakrishnan, R., Dral, P. O., Rupp, M., & Von Lilienfeld, O. A. (2014). Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 140022. <https://doi.org/10.1038/sdata.2014.22>
40. Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B. A., Thiessen, P. A., Yu, B., Zaslavsky, L., Zhang, J., Bolton, E. E. (2023). PubChem 2023 update. *Nucleic Acids Research*, 51, D1373–D1380. <https://doi.org/10.1093/nar/gkac956>
41. Ruddigkeit, L., Van Deursen, R., Blum, L. C., & Reymond, J.-L. (2012). Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *Journal of Chemical Information and Modeling*, 52, 2864–2875.
42. Wishart, D. S., Knox, C., Guo, A. C., Shrivastava, S., Hassanali, M., Stothard, P., Chang, Z., Woolsey, J. (2006). DrugBank: A comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Research*, 34, D668–D672.
43. Weininger, D. (1988). SMILES, a chemical language and information system: 1: Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28, 31–36. <https://doi.org/10.1021/ci00057a005>
44. Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., Aspuru-Guzik, A. (2018). Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4, 268–276. <https://doi.org/10.1021/acscentsci.7b00572>
45. Olivcrona, M., Blaschke, T., Engkvist, O., & Chen, H. (2017). Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, 9, 1–14. <https://doi.org/10.1186/s13321-017-0235-x>
46. Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2001). Introduction to automata theory, languages, and computation, 2nd edition. *ACM SIGACT News*, 32, 60–65. <https://doi.org/10.1145/568438.568455>
47. Yoshikawa, N., Terayama, K., Sumita, M., Homma, T., Oono, K., Tsuda, K. (2018). Population-based de novo molecule generation, using grammatical evolution. *Chemistry Letters*, 47, 1431–1434. <https://doi.org/10.1246/cl.180665>
48. Knuth, D. E. (1968). Semantics of context-free languages. *Mathematical Systems theory*, 2, 127–145.
49. Dai, H., Tian, Y., Dai, B., Skiena, S., Song, L. (2018). *Syntax-directed variational autoencoder for structured data*. 6th International Conference on Learning Representation ICLR 2018—Conf Track Proc.

50. Janz, D., Van Der Westhuizen, J., Paige, B., Paige, B., Kusner, M. J., Hernández-Lobato, J. M. (2018). *Learning a generative model for validity in complex discrete structures*. 6th International Conference Learning Representation ICLR 2018—Conf Track Proc.
51. O’Boyle, N. M., & Dalke, A. (2018). DeepSMILES: An adaptation of SMILES for use in machine-learning of chemical structures. *ChemRxiv*, 1–9.
52. Krenn, M., Häse, F., Nigam, A. K., Friederich, P., Aspuru-Guzik, A. (2020). Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1, 45024. <https://doi.org/10.1088/2632-2153/aba947>
53. Todeschini, R., & Consonni, V. (2008). *Handbook of molecular descriptors* (Vol.11). Wiley.
54. Faulon, J. L., & Bender, A. (2010). *Handbook of chemoinformatics algorithms*. CRC Press.
55. Leguy, J., Cauchy, T., Glavatskikh, M., Duval, B., Da Mota, B. (2020). EvoMol: A flexible and interpretable evolutionary algorithm for unbiased de novo molecular generation. *Journal of Cheminformatics*, 12, 1–19. <https://doi.org/10.1186/s13321-020-00458-z>
56. Polishchuk, P. (2020). CReM: Chemically reasonable mutations framework for structure generation. *Journal of Cheminformatics*, 12, 1–18. <https://doi.org/10.1186/s13321-020-00431-w>
57. Jensen, J. H. (2019). A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chemical Science*, 10, 3567–3572. <https://doi.org/10.1039/c8sc05372c>
58. Simonovsky, M., & Komodakis, N. (2017). *Dynamic edge-conditioned filters in convolutional neural networks on graphs*. Proceedings—30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, pp. 29–38.
59. Köbler, J., Schöning, U., & Torán, J. (1993). *The graph isomorphism Problem: Its structural complexity*. Springer Science & Business Media.
60. Gebauer, N. W. A., Gastegger, M., & Schütt, K. T. (2019). *Symmetry-adapted generation of 3D point sets for the targeted discovery of molecules*. Advances in Neural Information Processing Systems, 32.
61. Simm, G. N. C., Pinsler, R., Hernández-Lobato, J. M. (2020). Reinforcement learning for molecular design guided by quantum mechanics. In A. Singh (Eds.), *37th International Conference on Machine Learning, ICML 2020* (pp. 8906–8916). PMLR.
62. Hawkins, P. C. D., Skillman, A. G., Warren, G. L., Ellingson, B. A., Stahl, M. T. (2010). Conformer generation with OMEGA: Algorithm and validation using high quality structures from the protein databank and Cambridge structural database. *Journal of Chemical Information and Modeling*, 50, 572–584. <https://doi.org/10.1021/ci100031x>
63. Riniker, S. (2017). Molecular dynamics fingerprints (MDFP): Machine learning from MD data to predict free-energy differences. *Journal of Chemical Information and Modeling*, 57, 726–741. <https://doi.org/10.1021/acs.jcim.6b00778>
64. Simm, G. N. C., & Hernández-Lobato, J. M. (2020). *A generative model for molecular distance geometry*. 37th International Conference on Machine Learning ICML 2020 PartF16814, pp. 8896–8905.
65. Xu, M., Luo, S., Bengio, Y., Peng, J., Tang, J. (2021). *Learning neural generative dynamics for molecular conformation generation*. ICLR 2021—9th International Conference on Learning Representation.
66. Ganea, O. E., Pattanaik, L., Coley, C. W., Barzilay, R., Jensen, K. F., Green, W. H., Jaakkola, T. S. (2021). GEOMOL: Torsional Geometric Generation of Molecular 3D Conformer Ensembles. *Advances in Neural Information Processing Systems*, 17, 13757–13769.
67. Axelrod, S., & Gómez-Bombarelli, R. (2022). GEOM, energy-annotated molecular conformations for property prediction and molecular generation. *Sci Data*, 9, 185. <https://doi.org/10.1038/s41597-022-01288-4>
68. Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65, 386–408. <https://doi.org/10.1037/h0042519>
69. Mikolov, T., Karafiat, M., Burget L, Jan, C., Khudanpur, S. (2010). *Recurrent neural network based language model*. Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010. Makuhari, pp. 1045–1048.

70. Pérez-Ortiz, J. A., Gers, F. A., Eck, D., & Schmidhuber, J. U. (2003). Kalman filters improve LSTM network performance in problems unsolvable by traditional recurrent nets. *Neural Networks*, 16, 241–250. [https://doi.org/10.1016/S0893-6080\(02\)00219-8](https://doi.org/10.1016/S0893-6080(02)00219-8)
71. Sutskever, I., Martens, J., & Hinton, G. (2011). *Generating text with recurrent neural networks*. Proceedings of the 28th International Conference on Machine Learning, ICML 2011, pp. 1017–1024.
72. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9, 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
73. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014). *Learning phrase representations using RNN encoder-decoder for statistical machine translation*. EMNLP 2014—2014 Conference on Empirical Methods Natural Language Process Proc Conf, pp. 1724–1734. <https://doi.org/10.3115/v1/d14-1179>
74. Chollet, F. (2018). *Deep learning with Python*. Shelter Island New York Manning Publ Co.
75. Kingma, D. P., & Welling, M. (2014). *Auto-encoding variational bayes*. 2nd International Conference Learning Representation ICLR 2014—Conf Track Proc.
76. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B. (2015). *Adversarial autoencoders*. arXiv Prepr arXiv151105644
77. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63, 139–144.
78. Mirza, M., & Osindero, S. (2014). *Conditional generative adversarial nets*. arXiv Prepr arXiv14111784
79. Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., Liu, C. (2018). *A survey on deep transfer learning*. Artificial Neural Networks and Machine Learning—ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4–7, 2018, Proceedings, Part III 27. Springer, pp. 270–279.
80. Houlsby, N., Huszár, F., Ghahramani, Z., & Lengyel, M. (2011). *Bayesian active learning for classification and preference learning*. arXiv Prepr arXiv11125745.
81. Rogers, D., & Hahn, M. (2010). Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50, 742–754.
82. Consonni, V., & Todeschini, R. (2009). *Molecular descriptors for chemoinformatics: Volume I: alphabetical listing/VOLUME II: appendices, references*. Wiley.
83. Sabando, M. V., Ponzoni, I., Milios, E. E., & Soto, A. J. (2022). Using molecular embeddings in QSAR modeling: Does it make a difference? *Briefings in Bioinformatics*, 23, bbab365.
84. Chithrananda, S., Grand, G., & Ramsundar, B. (2020). *Chemberta: Large-scale self-supervised pretraining for molecular property prediction*. arXiv Prepr arXiv201009885
85. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A. (2017). Mastering the game of go without human knowledge. *Nature*, 550, 354–359.
86. Popova, M., Isayev, O., & Tropsha, A. (2018). Deep reinforcement learning for de novo drug design. *Science Advances*, 4, eaap7885.
87. Ertl, P., & Schuffenhauer, A. (2009). Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1, 1–11.
88. Wang, J., Hsieh, C.-Y., Wang, M., Wang, X., Wu, Z., Jiang, D., Liao, B., Zhang, X., Yang, B., He, Q. (2021). Multi-constraint molecular generation based on conditional transformer, knowledge distillation and reinforcement learning. *Nature Machine Intelligence*, 3, 914–922.
89. Gottipati, S. K., Sattarov, B., Niu, S., Pathak, Y., Wei, H., Liu, S., Blackburn, S., Thomas, K., Coley, C., Tang, J. (2020). *Learning to navigate the synthetically accessible chemical space using reinforcement learning*. International Conference on Machine Learning. PMLR, pp. 3668–3679.
90. Degen, J., Wegscheid-Gerlach, C., Zaliani, A., & Rarey, M. (2008). On the art of compiling and using “drug-like” chemical fragment spaces. *ChemMedChem*, 3, 1503–1507. <https://doi.org/10.1002/cmdc.200800178>

91. Bemis, G. W., & Murcko, M. A. (1996). The properties of known drugs. 1. *Molecular frameworks*. *Journal of Medicinal Chemistry*, 39, 2887–2893. <https://doi.org/10.1021/jm9602928>
92. Preuer, K., Renz, P., Unterthiner, T., Hochreiter, S., Klambauer, G. (2018). Fréchet ChemNet distance: A metric for generative models for molecules in drug discovery. *Journal of Chemical Information and Modeling*, 58, 1736–1741. <https://doi.org/10.1021/acs.jcim.8b00234>
93. Kosugi, T., & Ohue, M. (2021). Quantitative estimate index for early-stage screening of compounds targeting protein-protein interactions. *International Journal of Molecular Sciences*, 22, 10925.
94. Blaschke, T., Arús-Pous, J., Chen, H., Margreitter, C., Tyrchan, C., Engkvist, O., Papadopoulos, K., Patronov, A. (2020). REINVENT 2.0: An AI tool for de Novo drug design. *Journal of Chemical Information and Modeling*, 60, 5918–5922. <https://doi.org/10.1021/acs.jcim.0c00915>
95. Skalic, M., Sabbadin, D., Sattarov, B., Sciabolà, S., De Fabritiis, G. (2019). From target to drug: Generative modeling for the multimodal structure-based ligand design. *Molecular Pharmaceutics*, 16, 4282–4291. <https://doi.org/10.1021/acs.molpharmaceut.9b00634>
96. Ragoza, M., Masuda, T., & Koes, D. R. (2022). Generating 3D molecules conditional on receptor binding sites with deep generative models. *Chemical Science*, 13, 2701–2713. <https://doi.org/10.1039/d1sc05976a>
97. Wang, M., Hsieh, C.-Y., Wang, J., Wang, D., Weng, G., Shen, C., Yao, X., Bing, Z., Li, H., Cao, D., Hou, T. (2022). RELATION: A deep generative model for structure-based de novo drug design. *Journal of Medicinal Chemistry*, 65, 9478–9492. <https://doi.org/10.1021/acs.jmedchem.2c00732>
98. Wang, M., Wang, J., Weng, G., Kang, Y., Pan, P., Li, D., Deng, Y., Li, H., Hsieh, C. Y., Hou, T. (2022). ReMODE: A deep learning-based web server for target-specific drug design. *Journal of Cheminformatics*, 14, 84. <https://doi.org/10.1186/s13321-022-00665-w>
99. Hadfield, T. E., Imrie, F., Merritt, A., Birchall, K., Deane, C. M. (2021). Incorporating target-specific pharmacophoric information into deep generative models for fragment elaboration. *Journal of Chemical Information and Modeling*, 62, 2280–2292. <https://doi.org/10.1021/acs.jcim.1c01311>
100. Xie, W., Wang, F., Li, Y., Lai, L., Pei, J. (2022). Advances and challenges in de novo drug design using three-dimensional deep generative models. *Journal of Chemical Information and Modeling*, 62, 2269–2279. <https://doi.org/10.1021/acs.jcim.2c00042>
101. Zhang, J., & Chen, H. (2022). De novo molecule design using molecular generative models constrained by ligand-protein interactions. *Journal of Chemical Information and Modeling*, 62, 3291–3306. <https://doi.org/10.1021/acs.jcim.2c00177>
102. Ciepliński, T., Danel, T., Podlewska, S., & Jastrzębski, S. (2023). Generative models should at least be able to design molecules that dock well: A new benchmark. *Journal of Chemical Information and Modeling*, 63, 3238–3247. <https://doi.org/10.1021/acs.jcim.2c01355>
103. Guo, J., Janet, J. P., Bauer, M. R., Nittinger, E., Giblin, K. A., Papadopoulos, K., Voronov, A., Patronov, A., Engkvist, O., Margreitter, C. (2021). DockStream: A docking wrapper to enhance de novo molecular design. *Journal of Cheminformatics*, 13, 89. <https://doi.org/10.1186/s13321-021-00563-7>
104. García-Ortegón, M., Simm, G. N. C., Tripp, A. J., Hernández-Lobato, J. M., Bender, A., Bacallado, S. (2022). DOCKSTRING: Easy molecular docking yields better benchmarks for ligand design. *Journal of Chemical Information and Modeling*, 62, 3486–3502. <https://doi.org/10.1021/acs.jcim.1c01334>
105. Tan, X., Li, C., Yang, R., Zhao, S., Li, F., Li, X., Chen, L., Wan, X., Liu, X., Yang, T., Tong, X., Xu, T., Cui, R., Jiang, H., Zhang, S., Liu, H., Zheng, M. (2022). Discovery of pyrazolo[3,4-d] pyridazinone derivatives as selective DDR1 inhibitors via deep learning based design, synthesis, and biological evaluation. *Journal of Medicinal Chemistry*, 65, 103–119. <https://doi.org/10.1021/acs.jmedchem.1c01205>
106. Arús-Pous, J., Patronov, A., Bjerrum, E. J., Tyrchan, C., Reymond, J. L., Chen, H., Engkvist, O. (2020). SMILES-based deep generative scaffold decorator for de-novo drug design. *Journal of Cheminformatics*, 12, 1–18. <https://doi.org/10.1186/s13321-020-00441-8>

107. Li, Y., Zhang, L., Wang, Y., Zou, J., Yang, R., Luo, X., Wu, C., Yang, W., Tian, C., Xu, H. (2022). Generative deep learning enables the discovery of a potent and selective RIPK1 inhibitor. *Nature Communications*, 13, 6891.
108. Ballarotto, M., Willems, S., Stiller, T., Nawa, F., Marschner, J. A., Grisoni, F., Merk, D. (2023). De novo design of Nurr1 agonists via fragment-augmented generative deep learning in low-data regime. *Journal of Medicinal Chemistry*, 66, 8170–8177. <https://doi.org/10.1021/acs.jmedchem.3c00485>
109. Moret, M., Friedrich, L., Grisoni, F., Merk, D., Schneider, G. (2020). Generative molecular design in low data regimes. *Nature Machine Intelligence*, 2, 171–180. <https://doi.org/10.1038/s42256-020-0160-y>
110. Moret, M., Helmstädtter, M., Grisoni, F., Schneider, G., Merk, D. (2021). Beam search for automated design and scoring of novel ROR ligands with machine intelligence**. *Angewandte Chemie—International Edition*, 60, 19477–19482. <https://doi.org/10.1002/anie.202104405>
111. Renz, P., Van Rompaey, D., Wegner, J. K., Hochreiter, S., Klambauer, G. (2019). On failure modes in molecule generation and optimization. *Drug Discovery Today: Technologies*, 32–33, 55–63. <https://doi.org/10.1016/j.ddtec.2020.09.003>
112. Gao, W., Fu, T., Sun, J., & Coley, C. (2022). Sample efficiency matters: A benchmark for practical molecular optimization. *Advances in Neural Information Processing Systems*, 35, 21342–21357.
113. Schneider, P., Walters, W. P., Plowright, A. T., Sieroeka, N., Listgarten, J., Goodnow, R. A., Fisher, J., Jansen, J. M., Duca, J. S., Rush, T. S., Zentgraf, M., Hill, J. E., Krutoholow, E., Kohler, M., Blaney, J., Funatsu, K., Luebkemann, C., Schneider, G. (2020). Rethinking drug design in the artificial intelligence era. *Nature Reviews Drug Discovery*, 19, 353–364. <https://doi.org/10.1038/s41573-019-0050-3>

Privacy in Generative Models: Attacks and Defense Mechanisms



Maryam Azadmanesh, Behrouz Shahgholi Ghahfarokhi,
and Maede Ashouri Talouki

Abstract The high ability of generative models to generate synthetic samples with distribution similar to real data samples brings many benefits in various applications. However, one of the most major elements in the success of generative models is the data that is used to train these models, and preserving privacy of this data is necessary. However, various studies have shown that the high capacity of generative models leads to memorizing the details of the training data by these models, and different attacks have been conducted against generative models which infer information about training data from trained model. Also, many privacy-preserving mechanisms have been proposed to defend against these attacks. In this chapter, after introducing the topic, the privacy attacks against generative models and relevant defense mechanisms are discussed. In particular, the privacy attacks and related privacy preserving methods are categorized and discussed. Then, some challenges and future research directions are examined.

Keywords Privacy · Generative models · Generative adversarial networks · Membership inference attack · Inversion attack

1 Introduction

Nowadays, the high ability of generative models to generate synthetic samples with distribution similar to real data samples brings many benefits in various applications. Data scarcity, data imbalance and missing data for training diagnostic models in medical and industrial fields are sample problems that can be solved by generative

M. Azadmanesh · B. S. Ghahfarokhi (✉) · M. A. Talouki
Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran
e-mail: shahgholi@eng.ui.ac.ir

M. Azadmanesh
e-mail: m.azadmanesh@eng.ui.ac.ir

M. A. Talouki
e-mail: m.ashouri@eng.ui.ac.ir

models. Also, generative models are promising approaches for improving the quality of real data samples and removing noise in applications where capturing high-quality samples faces various limitations. In many fields, generative model-based classifiers, such as GAN-based classifiers, are used to improve the classification accuracy. However, one of the most major elements in the success of generative models is the data that is used to train these models. This data can be important from competitive, commercial and even security viewpoints due to its sensitive, unique, rarity and difficulty in collecting. Furthermore, data privacy protection is a legal requirement. So, the fundamental question that arises is whether a trained generative model can reveal information about individual records of the training data or not. The answer to this question is important because publishing the generative models to generate an infinite number of synthetic samples is considered as a privacy-preserving substitute for dataset sharing.

The goal of generative models is to learn an approximation of the real data distribution (p_{data}). To do this, the generative models estimate a model (p_g) explicitly or generate synthetic samples from the estimated model. The learning objective is to make p_g as near as possible to the real data distribution (p_{data}). But, in practice, p_{data} is not accessible and only a low number of samples are available to train the generative model (training samples), and the fundamental problem is to learn the real data distribution from the training samples. Figure 1 shows the high-level view of learning generative model.

Generative models are trained by performing the optimization process iteratively (a large number of iterations) on a limited number of training samples. The limited number of training samples makes it possible that these samples do not represent the whole distribution of the training data. On the other hand, many of these models, such as Generative Adversarial Networks (GANs), have high capacity for learning complex data distributions. But this high capacity may result in memorizing the details of the training data. If the generative model memorizes the details of individual samples rather than the population's features, it may leak information about the individual samples of the training dataset and tend to generate synthetic samples very similar to the training data. Various attacks have been conducted by exploiting the effect of training data on the parameters and output of the generative models, which leads to information leakage about the training data from the trained model.

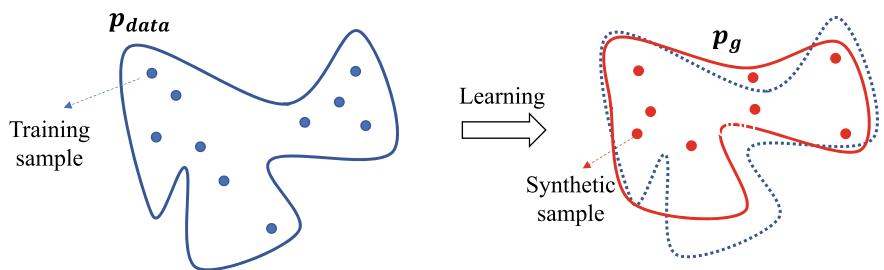


Fig. 1 High-level view of learning a generative model

The successful implementation of these attacks leads to the concern of data owners, data holders and model developers and hinders the use of these models in many applications. Therefore, understanding these attacks, the success factors of them, and the knowledge used by the attacker to conduct the attacks, can help the data owners and the model developers to estimate the risk caused by the release of the model or synthetic samples generated by the model. Also, having knowledge about such attacks can help data owners to decide how to release the trained model.

While a large number of works discuss privacy attacks against generative models, many defensive approaches have also been developed to counter these attacks with the aim of preserving the privacy and utility of generative models. Each of the defensive approaches can be effective in different scenarios of generative model publishing, and have advantages and disadvantages. Understanding the implementation of the defense approaches, their utilization, and the advantages and disadvantages of them, can help model developers to choose privacy-preserving solutions for training generative models according to the requirements and applications.

One of the widely accepted classes of generative models is GANs which receives noticeable attentions in recent years. Due to their high capability to produce high-quality synthetic samples of significant resemblance to the complex high-dimensional real data, GANs are applied in many fields. In addition, because of their popularity, most of researches in privacy of generative models have focused on GANs. In this chapter, the attacks against the GAN networks are examined, and in particular, the types of attacks and the attack strategies are discussed. Moreover, a taxonomy of the privacy attacks against GANs is presented. Then the privacy-preserving mechanisms for GANs are examined, and the taxonomy of the privacy-preserving mechanisms are provided. Then, the advantages and disadvantages of the defensive approaches are discussed, and the challenges and future directions are explained.

This chapter is organized as follows. The next section introduces GANs and the related concepts used in this chapter. The privacy attack against GANs is reviewed in Sect. 3. The privacy-preserving mechanisms in GANs and the related concepts are discussed in Sect. 4. The summary is presented in Sect. 5.

2 Generative Adversarial Network

Generative Adversarial Networks are a type of generative models that have received noticeable attention in recent years due to their high capability to model high-dimensional complex real data distribution. The initial idea of GAN is proposed by Goodfellow et al. [1], which introduced the GAN architecture consisting of two neural networks, i.e., a generator, G and a discriminator, D . The generator generates synthetic samples that are identical to real ones. The discriminator network is a binary classifier that separates synthetic samples from the samples drawn from real data distribution. The generator and discriminator can be represented by any differentiable function such as multi-layer neural network. The generator learns a mapping from a

latent distribution, p_z to the real data distribution, p_{data} using its parameters, θ_G , and the discriminator learns a mapping from real data distribution, p_{data} to real numbers, which estimates the divergence between p_{data} and synthetic data distribution, using its parameters, θ_D . The learning process is an iterative process in which discriminator and generator optimize their loss functions, i.e., $L_D(\theta_D, \theta_G)$ and $L_G(\theta_D, \theta_G)$, respectively. The learning process is iterated until finding the optimal point where the generator produces the real data distribution and the discriminator cannot discriminate between real and synthetic samples. In GAN, $L_G(\theta_D, \theta_G) = -L_D(\theta_D, \theta_G)$. Figure 2 shows the GAN architecture.

Because $L_G(\theta_D, \theta_G) = -L_D(\theta_D, \theta_G)$, the sum of the generator's and the discriminator's cost functions is zero. Therefore, the GAN training can be represented as a two-player zero-sum minmax game between the discriminator and the generator where the game converges to a Nash equilibrium point. This equilibrium point is a state in which none of the players, i.e., the discriminator nor the generator, can improve their cost.

Formally, the GAN optimization objective is represented as follows [2]:

$$\min_{\theta_G} \max_{\theta_D} L(\theta_D, \theta_G) = \left(\begin{array}{l} E_{x \sim p_{data}} [f(D_{\theta_D}(x))] \\ + E_{z \sim p_{latent}} [f(-D_{\theta_D}(G_{\theta_G}(z)))] \end{array} \right) \quad (1)$$

where θ_G and θ_D represent the parameters of the generator (G) and the discriminator (D), respectively. f is a concave function. If $f(x) = -\log(1 + \exp(-x))$ and $f(x) = x$, then original GAN optimization objective [1] and Wasserstein GAN (WGAN) optimization objective [3] are obtained, respectively.

The local Nash equilibrium (θ_G^*, θ_D^*) of the game satisfies the following two conditions in its local neighborhood

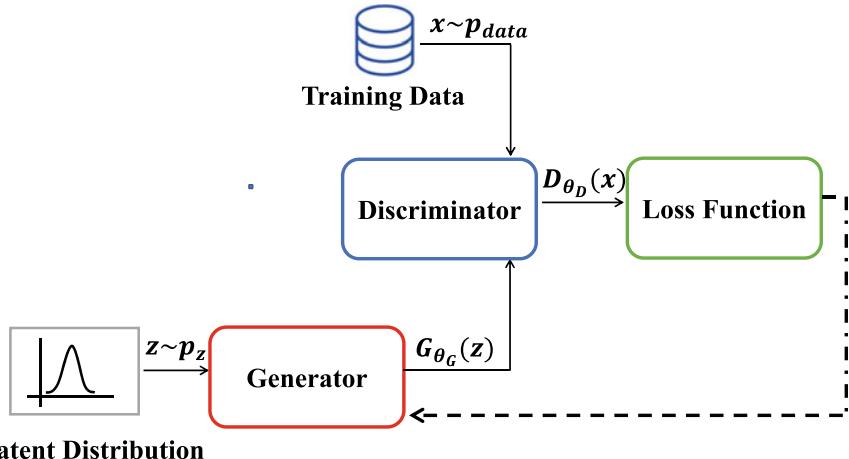


Fig. 2 GAN architecture

$$\theta_G^* = \operatorname{argmin}_{\theta_G} L(\theta_D^*, \theta_G), \text{ and } \theta_D^* = \operatorname{argmax}_{\theta_D} L(\theta_D, \theta_G^*). \quad (2)$$

As described, in GAN training, it is necessary to find a Nash equilibrium point, and in many problems, finding such an equilibrium point is a challenge. Therefore, in spite of the popularity of GANs in many applications, utilizing them in some problems has difficulties. One of the most important challenges of them is the instability problems such as vanishing gradients, non-convergence and mode collapse. To mitigate the convergence problems and improve the stability in the original GAN training, further studies propose new training procedures and architectures. These studies include changing the learning objective [3–5], introducing different regularization terms [6–8], and modifying the GAN architecture such as one generator/multiple discriminators [9–11] and one discriminator/multiple generators [12–14]. Since most of the proposed privacy attacks and defense techniques are independent of GAN versions, more details about the GAN improvement techniques are not provided in this section.

3 Privacy Attacks in Generative Models

Different attacks are conducted against Machine Learning (ML) models which are divided into two categories. In the first category, the target of the attack is the model itself, and attacker obtains information about the model or disrupts the model training or the model performance. This category of attacks includes model extraction, data poisoning, and evasion attacks. In the model extraction attack, the attacker infers information about the victim model and attempts to replicate the function of the model. Since in many applications, models are the proprietary information, this attack can lead to loss of revenue for the model developer. In data poisoning attack, the attacker tries to disrupt the model training and degrade the model performance by inserting malicious data into the training dataset. This attack reduces the trust in the model or the model developer. In the evasion attack, attacker provides malicious input to the trained model and leads to wrong inference or output of the model and performance degradation of the model. Unlike the attack of the first category targeting the ML model, the attacks of the second category aim to infer information about the training dataset. Model inversion attacks and membership inference attacks fall into this category. In model inversion attack (also known as property/attribute inference attack), the attacker obtains the values of the training records' attributes from the trained model's output. In the membership inference attack, the attacker determines if a target record is used in the model's training dataset or not.

Furthermore, from the perspective of the attack's execution time, attacks against ML models can be divided into two categories: attacks that are conducted during model training, and attacks that are conducted at the time of inference from the trained model. According to the attack procedure, some attacks must be conducted at a certain time. For example, since in the data poisoning attack, the attacker inserts

malicious data into the training dataset, this attack must be conducted during model training. On the contrary, some attacks can be conducted both during the model training and the model inference. For example, in membership inference attack, the attacker can determine whether the target record has been used to train the model or not at the inference time. Also, membership inference attacks can be implemented during training. For example, in federated learning, at the training time, each of the parties or the central server can find out whether a training record has been used to train the model by the other parties or not. The attacker's involvement in the attack can be active or passive. In the active attack, the attacker can actively change the training procedure in order to increase his/her gain, but in the passive attack, the attacker does not interfere in the training procedure. For example, when in the membership inference attack, the attacker accesses the trained model and determines whether the target record has been used in the training of the model or not, the attacker's involvement in the attack is passive. But if, like the attack presented in federated learning by Nasr et al. [15], the attacker changes the parameters of his/her local model in such a way that he/she can decide whether other parties use the target training record or not, the attacker's participation is active.

This section focuses on attacks that attempt to infer information about the training data records from the trained GAN models. In fact, the purpose of this section is to answer the question of what information about training data, the trained GAN model can leak. Answering this question will help data holders and model developers understand the privacy risk of publishing the trained GAN model. As described, such attacks are conducted during inference time to extract information about the training data. Because the attackers do not interfere in the training procedure, the attackers' involvements are passive, and the attacks compromise the confidentiality of the training data. Figure 3 shows the taxonomy of attacks against machine learning models and the red dashed rectangles indicate the category of the attacks presented in this section.

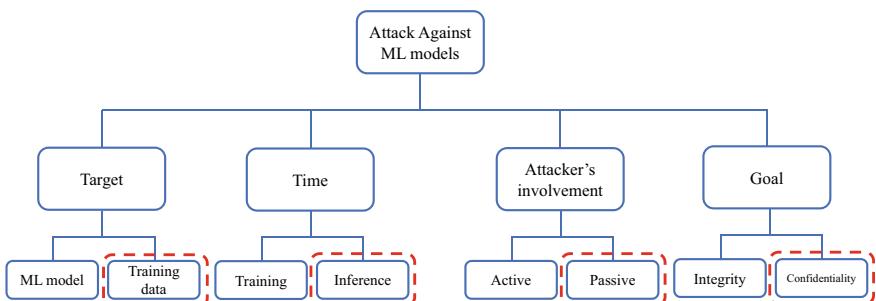


Fig. 3 Taxonomy of attacks against machine learning models, the red dashed rectangles indicate the type of attacks presented in this chapter

3.1 Membership Inference Attack

Membership inference attack is the most common privacy attack conducted against generative models. In membership inference attack, an attacker wants to determine whether a sample is used in the training data or not. Membership inference attack violates privacy. For example, if an insurance company has access to a set of synthetic records related to patients with stomach cancer and the company can identify that the health record of an applicant was used in the generation of the synthetic records, it will know the applicant's medical history, and the privacy of the applicant is violated.

Formally, given a target data, x_{target} , a trained generative model, $M(\theta)$, and prior knowledge auxiliary, aux , a membership inference attack is defined as the follows:

$$A : x_{target}, M(\theta), aux \rightarrow \{0, 1\} \quad (3)$$

where 0/1 output means x_{target} is a non-member/member of $M(\theta)$'s training data.

From the Bayesian point of view [16], in membership inference attack, the attacker computes the probability of x_{target} belonging to the training data, i.e. $P(x_{target} \in D_{train} | x_{target}, M(\theta), aux)$, and considers x_{target} as the training member, if the probability of $x_{target} \in D_{train}$ is higher than $x_{target} \notin D_{train}$. Formally,

$$A(x_{target}, M(\theta), aux) = \mathbb{I}\left[\log \frac{P(x_{target} \in D_{train} | x_{target}, M(\theta), aux)}{P(x_{target} \notin D_{train} | x_{target}, M(\theta), aux)} \geq 0\right] \quad (4)$$

where $\mathbb{I}[\cdot]$ is the indicator function, and D_{train} denotes the training dataset.

3.1.1 Adversarial Knowledge

Attacker can have different information about the generative model ($M(\theta)$) in Eqs. 3 and 4) and the training dataset (aux in Eqs. 3 and 4) and various attacks can be conducted based on this. The knowledge about the model includes model output, model architecture, model parameters and model training algorithm. In general, the attacker's access to learning models can be divided into two categories: white-box and black-box. In the black-box setting, the attacker accesses to the outputs of the model, and in the white-box setting, the model architecture and parameters are also accessible. In GAN, because it consists of two neural networks, a more detailed classification of the attacker's access to the model can be provided. According to the GAN architecture in Fig. 2, the types of attacker accesses to the GAN from the lowest level of knowledge to the highest level of knowledge are as follows [17]:

- 1. Access to synthetic samples.** In this case, the attacker only access to the output of the generator network, i.e., the generated synthetic samples. The number of synthetic samples available can be limited or unlimited.

2. **Access to $(z, G_{\theta_G}(z))$ pairs.** Under this setting, the attacker can access to the synthetic sample ($G_{\theta_G}(z)$) associated with the provided generator input (z). The number of pairs can be limited or unlimited.
3. **White-box access to the generator network.** In this setting, the attacker has white-box access to the generator network. Therefore, the input, the output, and the internal of the generator are available.
4. **Access to the discriminator network.** In this case, the attacker has access to the discriminator network, in addition to the generator network.

Each of the types of access can be related to different practical scenarios. In “Access to synthetic sample”, the developer can maintain ownership of the model and allow users to receive a number of synthetic samples, and in another scenario, the model developer publishes a number of generated synthetic samples. In “Access to $(z, G_{\theta_G}(z))$ pairs”, the developer allows the user to control the generated synthetic sample by changing the input of the generator network. In “White-box access to the generator”, the developer publishes the trained generator network. Therefore, the users can generate an unlimited number of synthetic samples. In “Access to the discriminator network”, the whole trained model is published. What is certain is that successful attacks with less knowledge about the model, highlight the possibility of more practical attacks, which are more dangerous than successful attacks with more knowledge. Figure 4 shows the membership inference attack with different knowledge levels about the model.

Knowledge about training data indicates the distribution of the training dataset. Given that the generative models approximate the training data distribution, most membership inference attacks against generative models assume that the training data distribution is approximately the same as the synthetic data distribution (i.e., $P_{G_{\theta_G}(Z)} \approx P_{D_{train}}$). These attacks do not consider any auxiliary knowledge about the training dataset. In contrast, other attacks assume that the training data distribution is available to the attacker. In fact, the attacker has access to the auxiliary dataset (D_{ref}), which is obtained by independent sampling from the training data distribution. Breugel et al. [18] show that if an auxiliary dataset is not available, the differences

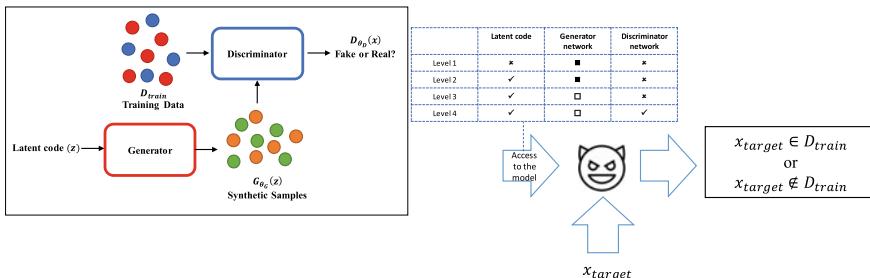


Fig. 4 Membership inference attack with different knowledge level about the model.(Level 1: access to synthetic data, Level 2: Access to $(z, G_{\theta_G}(z))$ pairs, Level 3: White-box access to the generator network, Level 4: Access to the discriminator network. ■: black-box access, and □: white-box access

between the training data distribution and synthetic data distribution can lead to misdiagnosis of overfitting and incorrect conclusion in membership inference attacks.

3.1.2 Membership Inference Attack Approaches

As Fig. 2 shows, in the training process of GAN network, in each step of training, a random batch of training data samples and a random batch of synthetic samples are injected as input to the discriminator, and the loss function is calculated. Then, the parameters of the discriminator are updated to reduce the loss function. In fact, the training data directly affects the parameters and the output of the discriminator. The connection between the generator and the training data is indirect and through the discriminator network. At each step of updating the generator parameters, information about the training data is propagated to the generator network through the gradient of the loss function with respect to the discriminator parameters. When GAN is overfitted to the training data, the parameters and output of the discriminator network, the generated synthetic samples, and the parameters of the generator network can be used to infer whether a target record has been used to train the target model. Different membership inference attacks used this information to conduct the attack. These attacks can be divided into three categories: (1) shadow model training methods, (2) distance calculation methods, and (3) density estimation methods.

Shadow model training methods. In this category, the output of the discriminator network in GAN is used to learn the statistical difference between the members and non-members of the training dataset. In these methods, synthetic samples generated by the target network are utilized to train a shadow GAN network (D' , G'), then the output of the discriminator (D') is used to infer whether the target record is a member of training dataset or not. Figure 5 shows a high-level Overview of shadow model training methods.

Hayes et al. [19] show that the discriminator of the target model gives higher confidence values to the members of the training dataset than non-members. Therefore, if the discriminator is accessible to the attacker, the output of the discriminator can be exploited by the attacker to conduct the attack. In the case where only generated synthetic samples are available, Hayes et al. [19] train a shadow model using the synthetic samples. Then, the output of the discriminator in the shadow model is used to conduct the attack. They also propose the utilization of the auxiliary dataset (D_{ref}) to increase the accuracy of the attack. Hu and Pang [20] propose a shadow model training method which is based on over-representation of GAN models. In the proposed method, a number of synthetic samples are generated by the victim model, and these samples are divided into two categories: training and test. The training set is used to train a shadow model. Then, the discriminator's outputs of the shadow model for training and test samples are used to cluster the samples into regions, and a score is assigned to each region based on the ratio of the number of training and test samples. To infer the membership of the target sample, the output of the discriminator network is used to determine the target sample's cluster. If the target sample belongs to a higher score region, the sample is predicted as a member. In the

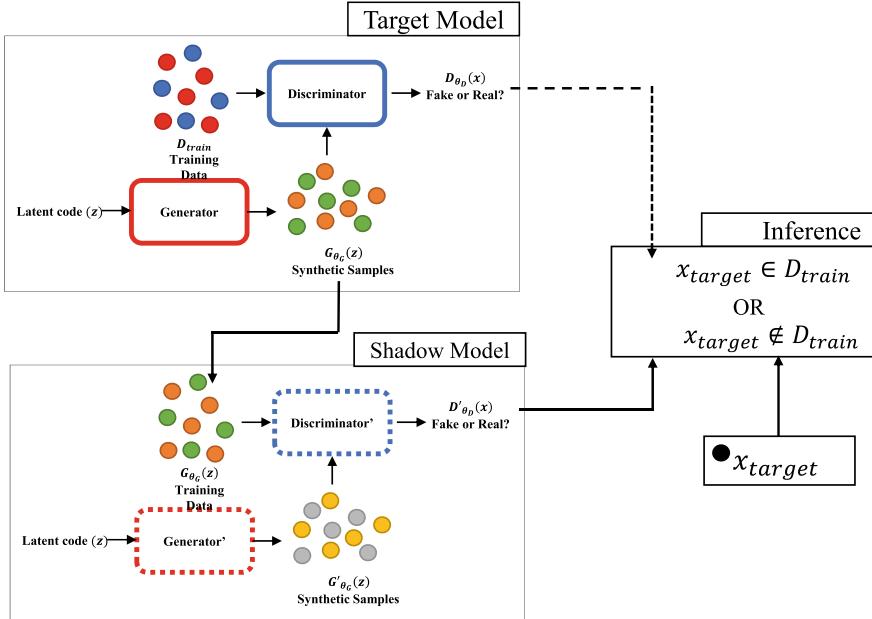


Fig. 5 A high-level overview of shadow model training methods

proposed method by Hu and Pang [20], it is assumed that the attacker has access to the discriminator of the victim network. In the presented methods in this category, most of the execution time in the attack is related to the shadow model training, and the inference about the membership of each target sample takes less time.

Distance calculation methods. In this category, by accessing the synthetic samples generated by the victim model, a measure (L) is calculated, which can be used to calculate the distance between the target sample and the synthetic samples and to infer the membership or non-membership of the target sample. This measure in different attacks includes the number of nearest synthetic samples to the target sample, the reconstruction error of the target sample, and the representation similarity between the target sample and the artificial samples. Figure 6 shows the high-level overview of distance calculation methods.

Hilprecht et al. [21] propose a membership inference attack with access to the generated synthetic samples. In the presented attack, a target sample with the greatest number of closest synthetic samples is predicted as a member of the training data. Also, Hilprecht et al. [21] proposed a set membership inference attack in which the attacker infers the membership or non-membership of a set of target samples. In this attack, for each target sample, the distance with all synthetic samples must be calculated. Chen et al. [17] propose a distance calculation method by computing the reconstruction error of the target sample. The proposed attack is implemented with different levels of access to the victim model, i.e., access to the synthetic samples, access to $(z, G_{\theta_G}(z))$ pairs, and white-box access to the generator network. In “access

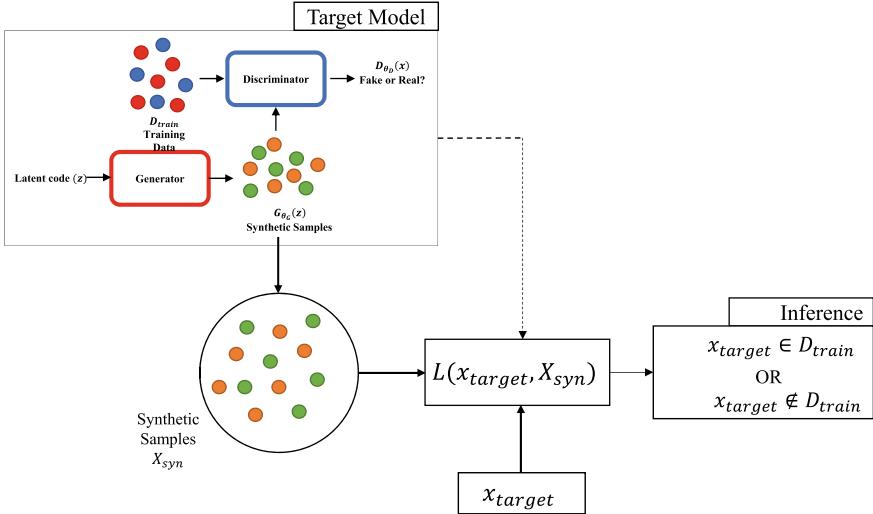


Fig. 6 A high-level overview of distance calculation methods

to $(z, G_{\theta_G}(z))$ pairs”, the attacker uses latent code (z) to find a better estimation of the target sample’s reconstruction error. In “white-box access to the generator network”, the attacker exploits the gradient information to solve the reconstruction problem more accurately. To improve the accuracy of the attack, they also present a calibration technique, which exploit an auxiliary dataset (D_{ref}). In the method proposed by Chen et al. [17], for each target sample, the problem of optimizing the reconstruction error must be solved. Liu et al. [22] present an attack, in which for each target sample, a neural network is trained to produce an input of the generator network such that the generator output matches the target sample. Then, the distance between the generator output and the target sample is used to infer about target sample’s membership. They also extend the attack to a set membership inference attack. Azadmanesh et al. [23] propose a membership inference attack with white-box access to the generator. In the attack, to calculate the reconstruction error, first, an auto-encoder is trained using generated synthetic samples, where the decoder has the same structure and parameters as the generator network, and the encoder is trained to map each input to the latent space (z). The learning objective is to reduce the reconstruction error of the synthetic samples. After training the auto-encoder, to infer the membership of each target sample, the sample is injected into the auto-encoder and its reconstruction error is obtained. If the reconstruction error of the target sample is low, it is considered as a member sample. In this method, most of the execution time in the attack is related to the auto-encoder training. Also, Zhang et al. [24] present a membership inference attack with access to the generated synthetic samples. In the proposed attack, first a representation of the synthetic samples is learned, then a measure is used to calculate the similarity between the representation of the target sample and the generated synthetic samples. A target sample is considered as a training member,

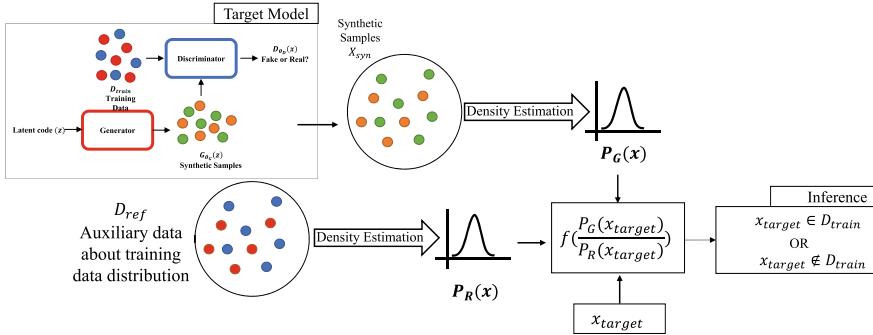


Fig. 7 A high-level overview of density estimation methods

if the similarity is greater than a certain threshold value. Zhang et al. also investigate the use of an auxiliary dataset to conduct the attack. In the presented attack, for each target sample, the similarity with all synthetic samples must be calculated.

Density estimation methods. Breugel et al. [18] propose a density estimation method. In the proposed method, the probability density of the generated synthetic samples ($P_G(x)$) is estimated by using the generated synthetic samples. Also, by accessing the auxiliary dataset (D_{ref}) that is sampled independently from the distribution of the training data, the attacker can estimate the probability density of the training data ($P_R(x)$). Then, based on the two probability densities of training data and synthetic data, the membership of each target sample is deduced. Figure 7 shows the high-level overview of the density estimation methods.

3.1.3 Taxonomy of the Membership Inference Attacks Against GANs

In order to provide an overview of proposed membership inference attacks, Fig. 8 shows the taxonomy of the membership inference attack against GANs. In this taxonomy, membership inference attacks are categorized based on knowledge about target victim model, knowledge about the training data, attack approach, attack execution time and number of target samples. In this classification, the attack execution time means whether most of the attack execution time is spent on training the attack model, and the inference about each sample using the attack model takes little time, or whether an optimization problem is required for the inference about each target sample. Also, the number of target samples means whether the attack model can infer the membership or non-membership of a set of target samples or only the membership or non-membership of a single target sample. Also, Table 1 shows a summary of papers about membership inference attacks against GAN networks. Compared to Fig. 8, this table provides more detailed information about each paper.

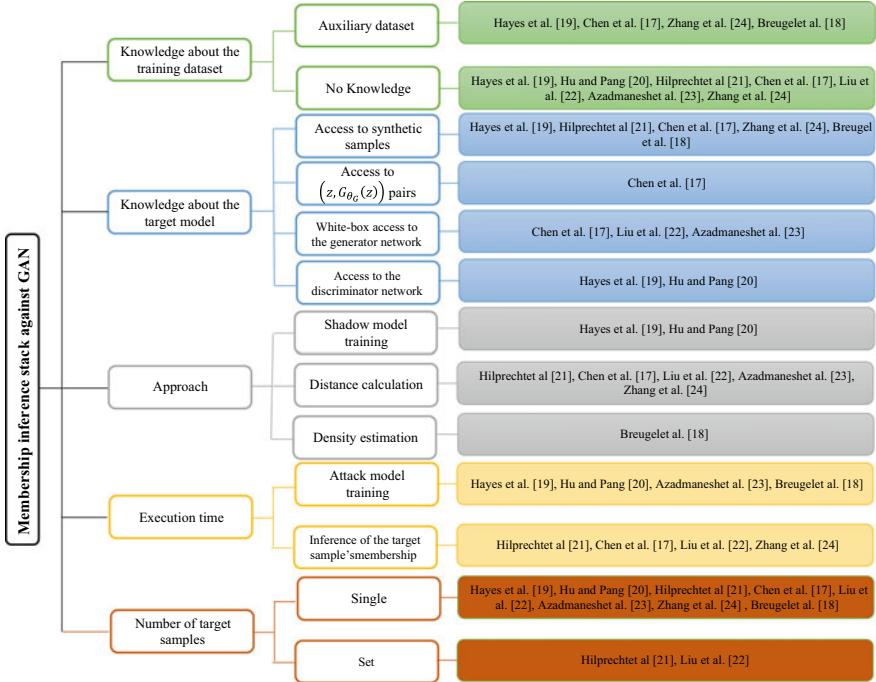


Fig. 8 Taxonomy of membership inference attacks against GANs

3.2 Model Inversion Attacks

In model inversion attack (also known as property/attribute inference attack), the sensitive attributes of the training data are reconstructed using their correlation with the output of the model. This attack violates privacy because it may reveal personal and sensitive information and unique features of the training dataset. Model inversion attacks are usually conducted against low-complexity machine learning models, and in particular, a limited number of model inversion attacks have been implemented against generative models. Zhou et al. [43] implement a property inference attack against GANs that extracts general property unrelated to the learning objective about the training data from the trained model. In this attack, first by accessing the model, a number of synthetic samples are generated. In the attack, two types of access are considered: access to synthetic samples and access to $(z, G_{\theta_G}(z))$ pairs. Then the artificial samples are fed to a classifier to label the samples with respect to the target property. Then, based on the outputs of the classifier, the information about the desired attribute is summarized. Webster et al. [44] conduct an identity inference attack in the face generation application, where the attacker can determine whether a sample with the same identity has been used for training or not. In the presented attack, a number of synthetic samples are generated by the model, and if the number

Table 1 Summary of top membership inference attacks against GANs

Refs.	Year	Knowledge about the training dataset	Knowledge about the target model	Approach	Execution time	# of target samples	Victim models	Dataset
[19]	2019	Yes/no	Access to synthetic samples, access to the discriminator network	Shadow model training	Attack model training	Single	DCGAN [25], BEGAN [26], DCGAN +VAE [27]	lfw [28], CIFAR-10 [29], Diabetic retinopathy [30]
[20]	2021	No	Access to the discriminator network	Shadow model training	Attack model training	Single	PGGAN [31], StyleGAN [32]	FFHQ [32]
[21]	2019	No	Access to synthetic samples	Distance calculation	Inference of the target's sample membership	Single/set	DCGAN [25]	MNIST [33], Fashion MNIST [34], CIFAR-10 [29]
[17]	2020	Yes/no	Access to synthetic samples/ access to $(z, G_{\theta_G}(z))$ pairs/ white-box access to the generator network	Distance calculation	Inference of the target's sample membership	Single	DCGAN [25], PGGAN [31], WGANGP [5], MedGAN [35], VAEGAN [36]	CelebA [37], MIMIC-III [38], Instagram New York [39]
[22]	2019	No	White-box access to the generator network	Distance calculation	Inference of the target's sample membership	Single/set	WGAN [3]	MNIST [33], CelebA [37], ChestX-ray8 [40]
[23]	2023	No	White-box access to the generator network	Distance calculation	Attack model training	Single	WGANGP [5], LSGAN [4], DRAGAN [6]	MNIST [33], Fashion MNIST [34], CelebA [37]

(continued)

Table 1 (continued)

Refs.	Year	Knowledge about the training dataset	Knowledge about the target model	Approach	Execution time	# of target samples	Victim models	Dataset
[24]	2022	Yes/no	Access to synthetic samples	Distance calculation	Inference of the target's sample membership	Single	–	–
[18]	2023	Yes	Access to synthetic samples	Density estimation	Attack model training	Single	GAN [1], WGANGP [5], CTGAN and TAVE [41]	California housing dataset [42], TVAE [41]

of images that have the same identity as the target image is greater than a threshold value, it is concluded that the image with the same identity has been used for training.

3.3 Future Directions

The main challenges in privacy attacks against GANs highlight future research directions. There are several research directions on privacy attacks for the future:

- The results of various attacks have shown that the more information about the victim's model is used in the attack, the higher the accuracy of the attack. Therefore, the developers should use privacy-providing mechanisms in the scenarios of publishing the trained GAN. On the contrary, in scenarios such as publishing the generated synthetic samples, the accuracy of the attack, when the number of model training samples is large, is low (as good as random guess). Lin et al. [45] prove that publishing a number of synthetic samples generated by a GAN satisfies some weak privacy guarantees. But in the presented proof, the imbalance of the training data is not considered. Although various attacks have been implemented in discriminative models against the underrepresented samples and the minority subsets of the training data [46–49], the implementation of practical attacks against these samples in GANs has been conducted in a limited way [18, 20]. Conducting of practical attacks against minority samples in GANs, especially in scenarios where the accuracy of the general privacy attacks is low, is one of the future research directions.
- One of the reasons for the success of the privacy attacks is the overfitting of the model to the training data. While in discriminative networks, overfitting can be measured using the generalization error, developing criteria to measure overfitting in the generative models is a challenge. One of the future directions is the development of a criteria to monitor the overfitting level of generative models or the development of methods for learning these models from the overfitting point of view.
- As Table 1 shows, different attacks have been implemented against different versions of GANs, the results of the attacks show that these attacks have different accuracies against different models. Most of the different versions of GAN have been developed to eliminate convergence problems and improve the stability in GAN training. A more in-depth analysis of the impact of these improvements on privacy leakage is a matter of future research.

4 Privacy-Preserving Mechanisms in GANs

In order to remove the privacy risk caused by publishing the GAN trained models or the generated synthetic samples, privacy mechanisms are combined with GANs. In other words, to prevent the privacy leakage of the training data by the parameters and output of the models, various methods change the training procedure of GANs. In general, these methods can be divided into two categories: (1) differential private defenses, (2) empirical defenses. In the first category [50–56], a rigorous mathematical privacy standard, i.e., differential privacy, is exploited in GANs. In the second category [57, 58], privacy is provided in such a way that the accuracy of a certain type of attack against GANs decreases. Furthermore, considering that there are different scenarios for the release of the trained model, in some methods [53–57] only the generator network is trained with the guarantee of the training data's privacy. In contrast, in other methods [50–52, 58], both the discriminator and the generator networks are trained with the privacy guarantees. From perspective of the generalizability, some methods [54] can only be applied to a certain type of GANs, while the idea of other methods [50–53, 55–58] can be generalized to different versions of GANs. To better understand the defensive methods, first, differential privacy and related concepts are reviewed in the following subsection, and then the privacy-preserving approaches are discussed.

4.1 Differential Privacy

Differential privacy is a rigorous mathematical privacy standard which provides a quantified measure of individual's privacy loss. “Adjacent databases” is a concept used in defining differential privacy. In ML application, adjacent databases indicate two training datasets with a difference in one record. Informally, a differential private algorithm is a randomized algorithm that its output is nearly identical on the adjacent datasets. Figure 9 shows the concept of differential privacy.

Formally, differential privacy is formulated as below:

Definition 1 $((\varepsilon, \delta)$ -differential privacy) [59]. A randomized mechanism, $F : \mathbb{X} \rightarrow \mathbb{Y}$, fulfills (ε, δ) -differential privacy if for any two adjacent inputs, $X, X' \in \mathbb{X}$, and for any subset of outputs, $S \subseteq \mathbb{Y}$,

$$\Pr[F(X) \in S] \leq e^\varepsilon \Pr[F(X') \in S] + \delta \quad (5)$$

Here, ε . and δ are referred to as privacy budget and confidence parameter, respectively. A lower ε . means a stronger privacy guarantee by F . To ensure the privacy of the individual records, $\delta < \frac{1}{\mathbb{X}_1}$., where \mathbb{X}_1 is the size of \mathbb{X} (Input of F .).

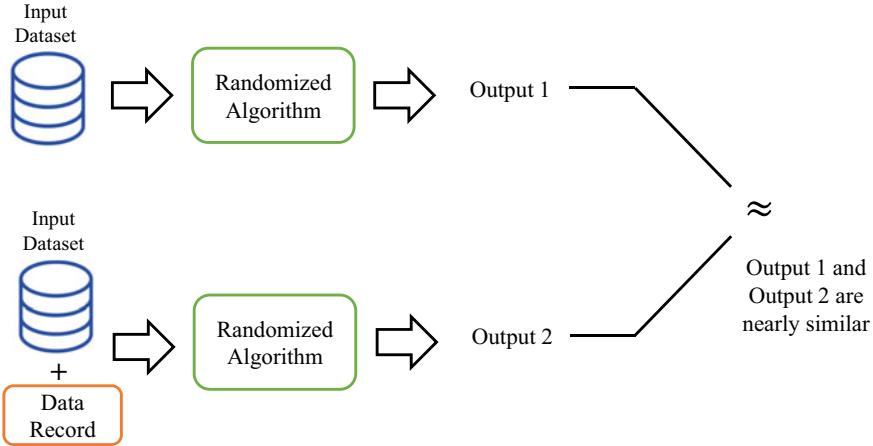


Fig. 9 The concept of differential privacy

One of the features of differential privacy is the post-processing feature, which means that any data-independent mapping of the output of differential private algorithm does not violate privacy. Formally, if $F : X \rightarrow Y$ fulfills (ϵ, δ) -differential privacy and $G : Y \rightarrow Y'$ is a data-independent function, then $G \circ F : X \rightarrow Y'$ also fulfills (ϵ, δ) -differential privacy [60].

Gaussian noise-based mechanism is a mechanism which provides differential privacy for a real-valued function by adding Gaussian noise scaled to the sensitivity of the function. The sensitivity of function f (i.e., S_f) is the maximum distance between its output for two adjacent inputs. Formally, S_f . is defined as.

$$S_f = \max_{\text{adjacent } X, X'} \|f(X) - f(X')\|, \quad (6)$$

where X and X' . are any two adjacent inputs.

Regarding sensitivity, S_f , Gaussian noise mechanism is formulated as.

$$F(X) \triangleq f(X) + N(0, \sigma^2), \quad (7)$$

where $N(0, \sigma^2)$ is a normal distribution with mean 0 and standard deviation σ . A Gaussian mechanism fulfills (ϵ, δ) -differential privacy if $\sigma \geq c S_f / \epsilon$ where the constant $c \geq \sqrt{2 \ln(1.25/\delta)}$ [60].

Another feature of differential privacy is composition, which permits the combination of multiple differential private mechanisms into one. It is clear that combining several differential private mechanisms leads to a greater loss of privacy, and the main issue is to determine the overall privacy loss.

To better understand the differential private defenses in GANs, it is appropriate to provide a high-level overview of applying the differential privacy to neural networks. In the neural network training procedure, at each step of training, a random batch

of training data is sampled, and the output and the loss function are calculated for these samples, then the gradients of the loss function with respect to the parameters are calculated, and based on the calculated gradients, the parameters of the neural networks are updated. If each step of training is replaced by a differential private algorithm, the calculated parameters in each step guarantee the privacy of the random batch of the training data. Clearly, the privacy of the samples of the training data, that are not present in the batch, also is preserved due to the lack of effect on the updated value of the parameters. Because the training procedure repeats several steps, the whole training process is actually a combination of several private differential mechanisms, and by using the combination theory, the amount of privacy loss of the final trained model can be calculated. Figure 10 shows a high-level overview of differential private learning in neural networks.

Various methods, including moments account method [61] and sample Gaussian mechanism [62], focus on providing a more accurate bound for privacy loss in composing Gaussian mechanisms with random sampling. Since the details of these methods are not necessary to understand the concepts of this chapter, more information is not provided in this section.

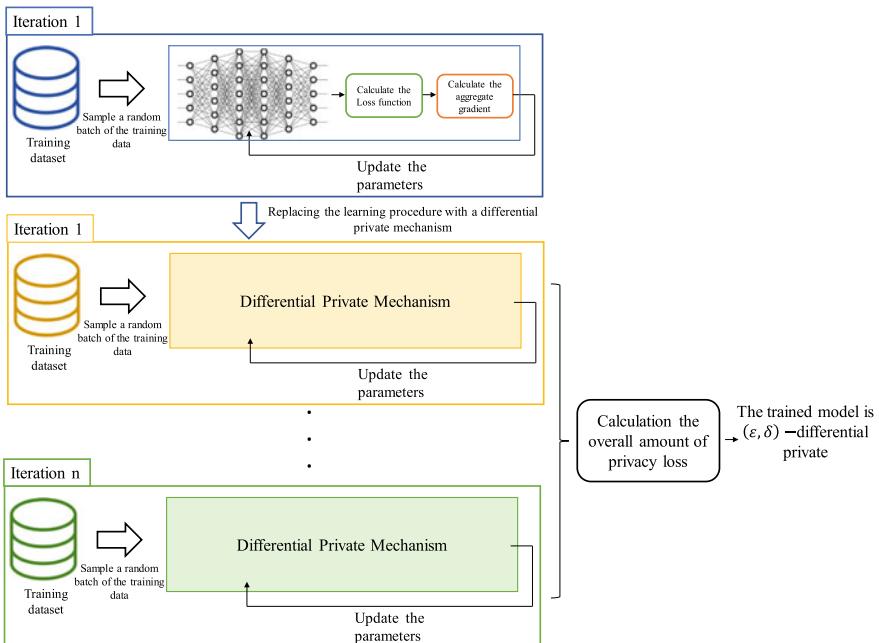


Fig. 10 High-level overview of differential private learning in neural networks

4.2 Privacy-Preserving Approaches in GANs

The privacy-preserving mechanisms provide the privacy of training data in different ways. These mechanisms are divided into three categories at the approach level: (1) methods that limit the effect of each training record on the gradients and the parameters of the discriminator network, (2) methods that limit the effect of each training record on the gradients and the parameters of the generator network, and (3) methods that use an adversarial classifier.

Limiting the effect of each training data record on the parameters and gradients of the discriminant network: In this category, the discriminator network is trained in a differential private manner. Because the connection between the generator and the training data is through the discriminator, based on the post-processing feature of differential privacy, the generator also guarantees the differential privacy of the training data. Different methods have been proposed to train the discriminator network in a differential private manner. Some method, such as [50], use the Gaussian mechanism to sanitize the gradients of the loss function with respect to the discriminator network parameters. Some methods, such as [51], use the Gaussian mechanism to sanitize the parameters of the discriminator network. Other methods such as [53] provide differential private input to the discriminator network.

Limiting the effect of each training data record on the parameters: Since in many scenarios only the generator network is published, in many methods, the generator network is trained in a differential private manner. As Eq. (1) shows, information about the training data is propagated from the discriminator to the generator through the loss function, or in other words, the gradients of the loss function with respect to the parameters of the generator network. Therefore, some methods, such [53], use the Gaussian mechanism to sanitize the cost function of the generator network. Also, some methods, such as [54–56], limit the effect of each training record on the gradient of the objective function with respect to the parameters of the generator network.

Using an adversarial classifier: In the methods of this category, the GAN is trained in a way that reduces the accuracy of membership inference attacks. For this purpose, an adversarial classifier is added to the GAN architecture. This classifier imitates the behavior of the attacker in the membership inference attack. In other words, in the learning process, the classifier tries to recognize the membership or non-membership of the records in the training data from the outputs of the generator, such as [57], or outputs of the discriminator network, such as [58]. On the other hand, the generator and the discriminator also try to reduce the classifier's gain.

4.3 Future Directions

There are several research directions on privacy-preserving mechanisms for the future:

- The results of the attacks against GAN models show that the smaller the number of training data samples, the higher the accuracy of the attacks. On the other hand, the utility of the model decreases when the numbers of training samples decrease. Furthermore, because when the number of training samples is small, each training sample has a greater effect on the model's parameters and output, in differential privacy-preserving mechanism, more noise is added to guarantee a certain level of privacy. Therefore, although methods such as [51] try to reduce the distortion of the parameters by determining appropriate noise, providing privacy-preserving mechanism that can preserve the utility of the model and the differential privacy of the training data when the number of training samples is small, is yet a challenge.
- Training data imbalance exists in most applications. Preliminary research has shown that the minority samples are more vulnerable to privacy attacks. Evaluation of defensive methods, especially empirical methods, in guaranteeing the privacy is a matter of future research. Furthermore, presenting privacy-preserving mechanisms considering the minority samples is another future direction.
- The preliminary research [63] proves that the greater the privacy guarantee, the greater the instability in convergence of GAN training. Investigating whether it is possible to change the learning objective in GANs in such a way that both privacy and convergence are preserved, is a matter of future research.

5 Summary

In this chapter, first, a classification of attacks against machine learning models was presented, and then the attacks that compromise the privacy of training data in inference time are examined in detail. A classification of these attacks presented in terms of attacker's knowledge about target victim model and the training data, attack approach, attack execution time, and number of target samples. Then, defense strategies that try to reduce the accuracy of privacy attacks were analyzed and categorized. Also, challenges and research opportunities for both privacy attacks and defense solutions were discussed.

References

1. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farely, D., Ozair, S., Courville, A., & Bengio, Y. (2014). *Generative adversarial nets*. 27th International conference on neural information processing systems, pp. 2672–2680.

2. Nagarajan, V., & Kolter, J. Z. (2017). *Gradient descent GAN optimization is locally stable*. Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, pp. 5591–5600.
3. Arjovsky, M., Chintala, S., & Bottou, L. (2017). *Wasserstein generative adversarial networks*. International conference on machine learning, pp. 214–223.
4. Mao, M., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., & Smolley, S. P. (2017). *Least squares generative adversarial networks*. 2017 IEEE International Conference on Computer Vision, pp. 1–17.
5. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). *Improved training of Wasserstein GANs*. Annual conference on neural information processing systems (NIPS), pp. 5767–5777.
6. Kodali, N., Hays, J., Abernethy, J., & Kira, Z. (2018). *On convergence and stability of GANs*. ICLR 2018 Conference Blind Submission, pp. 1–18
7. Sønderby, C. K., Caballero, J., Theis, L., Shi, W., & Huszar, F. (2017). *Amortized MAP inference for image superresolution*. [arXiv:1610.04490](https://arxiv.org/abs/1610.04490).
8. Roth, K., Lucchi, A., Nowozin, S., & Hofmann, T. (2017). *Stabilizing training of generative adversarial networks through regularization*. Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, pp. 2015–2025.
9. Jin, Y., Wang, Y., Long, M., Wang, J., Philip, S. Y., & Sun, J. (2020). *A multi-player minimax game for generative adversarial networks*. Proceedings IEEE International Conference on Multimedia and Expo (ICME), pp 1–6.
10. Hardy, C., Le Merrer, E., & Sericola, B. (2019). *MD-GAN: Multi-discriminator generative adversarial networks for distributed datasets*. Proceedings IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 866–877.
11. Nguyen, T., Le, T., Vu, H., & Phung, D. (2017). *Dual discriminator generative adversarial nets*. Proceedings advances in neural information processing systems (NIPS), pp. 2670–2680.
12. Hoang, Q., Nguyen, TD, Le, T., & Phung, D. (2018). *MGAN: training generative adversarial nets with multiple generators*. Proceedings International Conference on Learning Representations.
13. Ke, S., & Liu, W. (2020). Consistency of multiagent distributed generative adversarial networks. *IEEE Transactions on Cybernetics*, 1, 1–11.
14. Ghosh, A., Kulharia, V., Namboodiri, VP, Torr, P., Dokania, P. (2018). *Multi-agent diverse generative adversarial networks*. Proceedings IEEE Conference on Computer Vision and Pattern Recognition, pp. 8513–8521.
15. Nasr, M., Shokri, R., & Houmansadr, A. (2019). *Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning*. 2019 IEEE symposium on security and privacy (S&P). IEEE, pp. 739–753.
16. Sablayrolles, A., Douze, M., Schmid, C., Ollivier, Y., & Jégou, H. (2019). *White-box vs black-box: Bayes optimal strategies for membership inference*. International Conference on Machine Learning (ICML). JMLR, pp. 5558–5567.
17. Chen, D., Yu, N., Zhang, Y., & Fritz, M. (2020). *GAN-leaks: A taxonomy of membership inference attacks against generative models*. The 2020 ACM SIGSAC conference on computer and communications security, pp. 343–362.
18. Breugel, B. V., Sun, H., Qian, Z., & Schaer, M. (2023). *Membership inference attacks against synthetic data through overfitting detection*. [arXiv:2302.12580](https://arxiv.org/abs/2302.12580)
19. Hayes, J., Melis, L., Denerzis, G., & De Cristofaro, E. (2019). *LOGAN: membership inference attacks against generative models*. Privacy enhancing technologies symposium, pp. 133–152.
20. Hu, H., & Pang, J. (2021). *Membership inference attacks against GANs by leveraging over-representation regions*. Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, pp. 2387–2389.
21. Hilprecht, B., Harterich, M., & Bernau, D. (2019). *Monte Carlo and reconstruction membership inference attacks against generative models*. Privacy enhancing technologies symposium, pp. 232–249.

22. Liu, K. S., Xiao, C., Li, B., & Gao, J. (2019). *Performing co-membership attacks against deep generative models*. 2019 IEEE International Conference on Data Mining (ICDM), pp. 459–467.
23. Azadmanesh, M., Ghahfarokhi, B. S., & Talouki, M. A. (2023). An auto-encoder based membership inference attack against generative adversarial network. *The ISC International Journal of Information Security*. <https://doi.org/10.22042/iscure.2022.320282.735>.
24. Zhang, Z., Yan, C., & Bradley, A. M. (2022). Membership inference attacks against synthetic health data. *Journal of Biomedical Informatics*, 125, 1–12.
25. Radford, A., Metz, L., & Chintala, S. (2015). *Unsupervised representation learning with deep convolutional generative adversarial networks*. [arXiv:1511.06434](https://arxiv.org/abs/1511.06434)
26. Berthelot, D., Schumm, T., & Metz, L. (2017). BEGAN: Boundary equilibrium generative adversarial networks. [arXiv:1703.10717](https://arxiv.org/abs/1703.10717)
27. Larsen, A. B. L., Sønderby, S. K., Larochelle, H., & Winther, O. (2016). Autoencoding beyond pixels using a learned similarity metric. In *ICLM* (pp 1–8)
28. Huang, G. B., Ramesh, M., Berg, T., Miller, E. L. (2007). *Labeled faces in the wild: A database for studying face recognition in unconstrained environments*. Technical report. <http://vis-www.cs.umass.edu/lfw/lfw.pdf>
29. Krizhevsky, A., & Hinton, G. (2009). *Learning multiple layers of features from tiny images*. Technical report. <https://www.cs.toronto.edu/%E2%88%BCkriz/learning-features-2009-TR.pdf>
30. Kaggle.com. (2015). *Diabetic retinopathy detection*. <https://www.kaggle.com/c/diabetic-retinopathy-detection#references>
31. Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2018). *Progressive growing of GANs for quality, stability, and variation*. Proceedings of International Conference on Learning Representations (ICLR).
32. Karras, T., Laine, S., & Aila, T. (2019). *A style-based generator architecture for generative adversarial networks*. Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4401–4410.
33. LeCun, Y., Cortes, C., & Burges, C. J. C. (1998). *The MNIST database of handwritten digits*. <http://yann.lecun.com/exdb/mnist/>
34. Xiao, H., Rasul, K., & Vollgraf, R. (2017). *Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms*. [arXiv:1708.07747](https://arxiv.org/abs/1708.07747)
35. Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W., & Sun, J. (2017). Generating multi-label discrete patient records using generative adversarial networks. *Proceedings of Machine Learning Research*, 68, 286–305.
36. Bhattacharyya, A., Fritz, M., & Schiele, B. (2019). “*Best-of-many samples*” distribution matching. [arXiv:1909.12598](https://arxiv.org/abs/1909.12598)
37. Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). *Face attributes in the wild*. IEEE International Conference on Computer Vision (ICCV), pp. 3730–3738.
38. Johnson, A. E. W., Pollard, T. J., Shen, L., Li-wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L., & Mark, R. (2016). MIMIC-III, A freely accessible critical care database. *Scientific Data*, 3, 160035. <https://doi.org/10.1038/sdata.2016.35>
39. Backes, M., Humbert, M., Pang, J., & Zhang, Y. (2017). *walk2friends: Inferring social links from mobility profiles*. ACM SIGSAC Conference on Computer and Communications Security (CCS), pp. 1943–1957.
40. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., & Summers, R. M. (2017). *Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2097–2106.
41. Xu, L., Skoulikidou, M., Cuesta-Infante, A., & Veeramachaneni, K. (2019). *Modeling tabular data using conditional GAN*. Proceedings of the 33rd International Conference on Neural Information Processing Systems, no. 659, pp. 7335–7345.
42. Pace, R. K., & Barry, R. (1997). Sparse spatial autoregressions. *Statistics & Probability Letters*, 33, 291–297.

43. Zhou, J., Chen, Y., Shen, C., & Zhang, Y. (2021). *Property inference attacks against GANs*. [arXiv:2111.07608](https://arxiv.org/abs/2111.07608)
44. Webster, R., Rabin, J., Simon, L., & Jurie, F. (2021). *This person (probably) exists. Identity membership attacks against GAN generated faces*. [arXiv:2107.06018](https://arxiv.org/abs/2107.06018).
45. Lin, Z., Sekar, Y., & Fanti, G. (2021). *On the privacy properties of GAN-generated samples*. Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, vol. 130, pp. 1522–1530.
46. Long, Y., Bindschaedler, V., Wang, L., Bu, D., Wang, X., Tang, H., Gunter, C. A., & Chen, K. (2018). *Understanding membership inferences on well-generalized learning models*. [arXiv: 1802.04889](https://arxiv.org/abs/1802.04889)
47. Long, Y., Wang, L., Bu, D., Bindschaedler, V., Wang, X., Tang, H., Gunter, C. A., Chen, K. (2020). *A pragmatic approach to membership inferences on machine learning models*. Proceedings of the 2020 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 521–534.
48. Yaghini, M., Kulynych, B., & Troncoso, C. (2019). *Disparate vulnerability: On the unfairness of privacy attacks against machine learning*. [arXiv:1906.00389](https://arxiv.org/abs/1906.00389)
49. Moore, H. D., Stephens, A., & Scherer, W. (2022). An understanding of the vulnerability of datasets to disparate membership inference attacks. *Journal of Cybersecurity and Privacy*, 2(4), 882–906.
50. Xu, C., Ren, J., Zhang, D., Zhang, Y., Qin, Z., & Ren, K. (2019). GANobfuscator: Mitigating information leakage under GAN via differential privacy. *IEEE Transactions on Information Forensics and Security*, 14(9), 2358–2371.
51. Azadmanesh, M., Shahgholi Ghahfarokhi, B., & Ashouri Talouki, M. (2023). ADAM-DPGAN: A differential private mechanism for generative adversarial network. *Applied Intelligence*, 53, 11142–11161.
52. Jordon, J., Yoon, J., & Schaar, M. (2019). *PATE-GAN: Generative synthetic data with differential privacy guarantees*. Seventh International Conference on Learning Representations, pp. 1–21.
53. Han, C., & Xue, R. (2021). Differentially private GANs by adding noise to discriminator's loss. *Computer and Security*, 107, 1–14.
54. Chen, D., Orekondy, T., & Fritz, M. (2020). *GS-WGAN: A gradient sanitized approach for learning differentially private generators*. 34 Conference on Neural Information Processing Systems, pp. 1–18.
55. Long, Y., Wang, B., Yang, Z., Kailkhura, B., Zhang, A., Gunter, C., & Li, B. (2021). Scalable differentially private generative student model via PATE. *Advances in Neural Information Processing Systems*, 34, 1–18 (NeurIPS 2021).
56. Wnag, B., Wu, F., Long, Y., Rimanic, L., Zhang, C., & Li, B. (2021). *DataLens: Scalable privacy preserving training via gradient compression and aggregation*. Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, pp. 2146–2168.
57. Mukherjee, S., Xu, Y., Trivedi, A., & Ferres, J. L. (2019). *PrivGan: Protecting GANs from membership inference attack at low cost*. [arXiv:2001.00071](https://arxiv.org/abs/2001.00071)
58. Ha, H., Hwang, U., Jang, J., Bae, H., & Yoon, S. (2022). *Membership Privacy-preserving GAN*. <https://bmvc2022 mpi-inf.mpg.de/0576.pdf>
59. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M. (2006). *Our data, ourselves: Privacy via distributed noise generation*. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 486–503.
60. Dwork, C., & Roth, A. (2013). The algorithmic foundations of differential privacy. *Theoretical Computer Science*, 9(3), 211–407.
61. Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). *Deep learning with differential privacy*. 2016 ACM SIGSAC Conference on computer and communications security, pp. 308–318.
62. Mironov, I., Talwar, K., & Zhang, L. (2019). *Renyi differential privacy of the sampled Gaussian mechanism*. [arXiv:1908.10530](https://arxiv.org/abs/1908.10530)

63. Azadmanesh, M., Shahgholi Ghahfarokhi, B., Ashouri Talouki, M., & Eliasi, H. (2023). On the local convergence of GANs with differential privacy: Gradient clipping and noise perturbation. *Expert System with Applications*, 224, 1–15.

Generative Adversarial Network for Synthetic Image Generation Method: Review, Analysis, and Perspective



Christine Dewi

Abstract Recently, generative adversarial networks (GANs) have been investigated since 2014, and many algorithmic solutions have been suggested for them. GAN have recently become a popular research topic. Even so, not many studies are deep enough to explain the relationship between the many variations of GAN and how they arise. We aim to provide a survey of different GAN techniques in this work, discussing them from the angles of theory, algorithms, and practical applications. We begin with a comprehensive introduction, architecture, and applications of the most popular GAN algorithms, then we draw parallels and draw distinctions between them. The second part of the study focuses on examining the theoretical issues associated with GANs. In this work, we try to determine the benefits and drawbacks of GANs, as well as the important obstacles that stand in the way of achieving successful implementation of GAN in a variety of application domains. Typical GAN applications are presented, including those in synthetic image processing and computer vision, natural language processing, music, speech, and audio, medicine, and data science. The final section of the research presents the study's conclusion and some suggestions for further research. Further, highlighting the pros and cons of ongoing studies on the application of aversive learning can help guide future research efforts toward the most fruitful avenues.

Keywords GAN · WGAN · DCGAN · Synthetic image · Traffic sign · Deep learning

1 Introduction

GANs have been around ever since their debut a few years ago. Mirza [53] have established themselves as an effective method for generative modeling that makes use of deep learning. GANs were first presented to the public in 2014 and were the

C. Dewi (✉)

Department of Information Technology, Satya Wacana Christian University, 52-60 Diponegoro Rd, Salatiga City 50711, Indonesia
e-mail: christine.dewi@uksw.edu

product of research on adversarial attacks that were carried out on neural networks [90]. GANs have exhibited the ability to generate new data that is pseudo-real, of high quality, and that replicates their training set. This is accomplished by applying adversarial attacks, which are otherwise damaging for neural networks. Deep fakes are now commonplace because of the proliferation of learning models like this one.

Deep learning algorithms are utilized in unsupervised learning, specifically for the purpose of producing new data samples that are like training data. GAN is one sort of deep learning algorithm. The generator and the discriminator are both types of neural networks that are included in a GAN's construction. These networks engage in a zero-sum game with each other and compete against one another. Both the generator network, which creates new data samples based on random input noise, and the discriminator network, which attempts to differentiate the created data from the actual data, are part of an artificial intelligence system. Training involves switching between the two networks so that both can improve their efficiency. A data generator attempts to make more realistic data in the hopes of tricking a data discriminator that is always improving its capacity to tell the difference between genuine and manufactured data. As the training continues, the generator will eventually learn to generate data that is more realistic than the discriminator is expecting, which will allow it to fool it. In turn, the discriminator improves its ability to tell the difference between actual data and produced data. GANs are a useful tool for data augmentation, image and video synthesis, and other applications in computer vision and natural language processing, as they can make new data samples that are almost indistinguishable from the real data. As a result, they can produce a generator that can produce new data samples.

Synthetic image generation is the process of using algorithms to create new images that do not exist in the real world [17, 34]. This process is often used in computer vision applications where the available data is limited or the cost of obtaining real images is high. Using a GAN is a common approach to creating synthetic images. Furthermore, GANs are a type of neural network architecture consisting of two independent sub-networks. While the generator creates new images from scratch using noise inputs, the discriminator evaluates their authenticity. With each iteration, the generator improves its ability to trick the discriminator with more convincing fake images. Another method for synthetic image generation is with Variational Autoencoders (VAEs). VAEs are another type of neural network that can generate new images by sampling from a learned latent space. This method can be used to generate images that are like training data, but with some variations or mutations. There are many applications of synthetic image generation, including data augmentation, creating images for training machine learning models, and generating realistic images of products for e-commerce. Synthetic image generation has also been used in artistic applications, such as creating new styles of art or generating new textures and patterns.

There are many types of GANs that have been developed since their introduction in 2014 [67]. Here are some examples of the most popular types of GANs: (1) Vanilla GANs: These are the simplest type of GANs that consist of only a generator and discriminator. They can be used to generate a wide variety of images but may suffer

from instability during training [89]. (2) Deep Convolutional GANs (DCGANs): These GANs use convolutional neural networks to generate images, which makes them more effective at handling images with complex features such as edges and textures. They also use techniques such as batch normalization and convolutional transpose to improve stability during training [24]. (3) Conditional GANs: These GANs generate images based on a specific input, such as an image or text description. This allows for more precise control over the output of the generator [65]. (4) CycleGANs: These GANs are designed for image-to-image translation tasks, such as transforming images from one domain to another, without the need for paired data for training. CycleGANs make advantage of cycle consistency loss to ensure that the output images match the input images exactly [69]. (5) Wasserstein GANs (WGANs): These GANs are more stable during training because they employ a novel loss function. They also use weight clipping or gradient penalty techniques to enforce a Lipschitz constraint on the discriminator [4]. (6) Progressive GANs: These GANs generate images of increasing resolution in a progressive manner, starting from low-resolution images and gradually adding more detail [82]. This allows for higher-quality images to be generated but requires a longer training time and more computational resources. These are just a few examples of the many types of GANs that exist. Each type has its own strengths and weaknesses and is suited for different applications [45].

This study aims to contribute to the clarification and classification of previous studies, which will, in turn, support readers, researchers, and practitioners who are interested in exploring this field of research. The following outline depicts the structure of the present document. Before continuing the research that has already been published in the relevant literature, Sect. 2 offers an introduction to the GAN models that are both currently accessible and actively in use. The following section, Sect. 3, provides a summary of the formula of each GAN model and discussions. A more comprehensive insight, literature review and discussion into the various applications of GAN is finally provided in Sect. 4. Finally, Sect. 5 exhibits the conclusions and prospective considerations.

2 Generative Adversarial Networks (GAN) Model

The architectural and topological limitations of the convolutional GAN were evaluated in 2016 by Alec Radford et al. DCGANs is a technology that is more reliable than previous methods [51, 59]. In a GAN, training on both the generating (G) and discriminative (D) components occurs simultaneously [5, 30]. Validity of data in a sample can be determined with the help of the discriminatory model. As a means of challenging the discriminative model, the generative model records a distribution of target data [73, 79]. The G model's training data is categorized as “realistic” or “unrealistic” by the D model, a binary classifier. By providing D with data that it may use to make a “real” classification, G can minimize its loss function, as depicted by Eq. (1).

$$\min \max(D, G) = E_{x-pdata(x)}[\log D(x)] + E_{x-p(x)}[\log(1 - D(G(z)))] \quad (1)$$

To enhance the general capabilities of GANs, a significant amount of research work has been focused on developing new GAN configurations. Implementing a GAN involves setting up the architecture of the generator and discriminator neural networks, defining the loss function, and setting up the training process. Here are some steps involved in GAN implementation: (1) Provide a description of the architecture of the neural networks that serve as the generator and discriminator: The generator network takes in noise at random as its input and produces images as its output. On the other hand, the discriminator network receives an image as its input and produces a probability that indicates whether the image is authentic. The design of the networks is determined, at least to some degree, by the GAN variation that is being used. (2) The term “loss function” refers to the tool that is utilized in the process of measuring the amount of deviation that exists between the generated images and the actual photos. The binary cross-entropy loss is used as a loss function in traditional GANs, while the Wasserstein distance is used in WGANs. (3) Establish the training procedure: The training procedure for the GAN consists of alternating between training the generator and discriminator networks. During training, the generator is given random noise to process, which causes it to produce fake images. In contrast, the discriminator is taught using a mix of genuine and fake images. The objective here is to teach the generator how to produce images that can fool the discriminator. (4) Once the GAN has been trained, the image generator can be used to generate new images by feeding it random noise after the GAN has been trained. (5) TensorFlow, PyTorch, and Keras are just a few examples of the many GAN libraries and frameworks that are available that can be used to simplify the implementation process. These frameworks come equipped with pre-built modules that can be utilized for constructing the networks, determining the loss function, and establishing the training procedure.

Between 2014 and 2018, the field of GANs saw significant developments and advancements in both the theory and applications of the model. Here are some notable events during this time as we can see in Fig. 1: (1) 2014: Ian Goodfellow and his colleagues introduce the GAN framework in their paper. (2) 2015: Alec Radford, Luke Metz, and Soumith Chintala introduce DCGAN (Deep Convolutional GAN), which uses convolutional neural networks (CNNs) in the generator and discriminator, leading to significant improvements in image generation quality. (3) 2016: Xudong Mao and his colleagues [59] introduce the concept of “mode collapse,” which occurs when the generator produces a limited set of outputs that do not fully capture the range of possible real data. This led to the development of various methods to address mode collapse in GANs, including InfoGAN [12] and WGAN. (4) 2017: Martin Arjovsky and his colleagues [8, 94] introduce WGAN, which uses the Wasserstein distance instead of the Jensen-Shannon divergence to measure the distance between the generator and discriminator distributions. This improves the stability of GAN training and leads to higher quality image generation. (5) 2018: NVIDIA researchers introduce StyleGAN [62], which uses a hierarchical generator architecture to control various aspects of image synthesis, such as style, color, and texture. This leads to

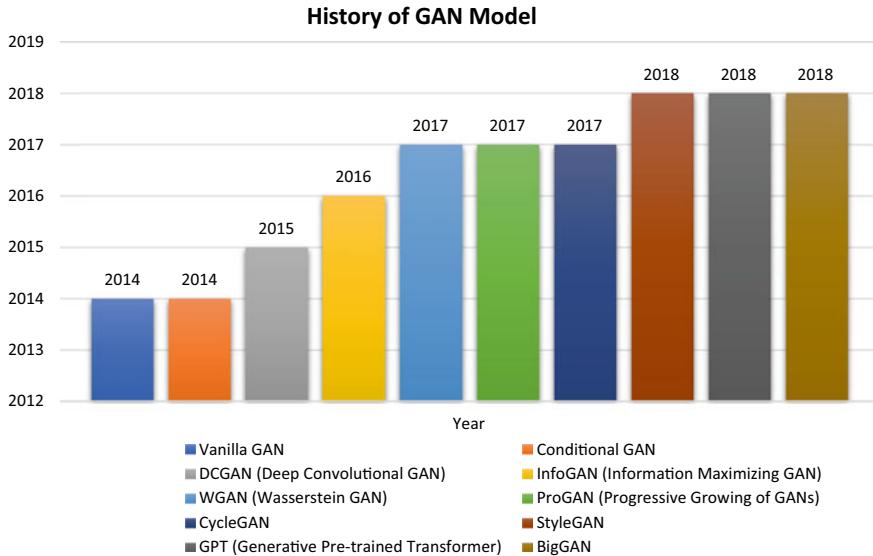


Fig. 1 History of GAN model

unprecedented quality and control over the generated images, and the method quickly becomes a popular approach in the GAN community.

Overall, the period between 2014 and 2018 saw significant advancements in GAN theory and applications, leading to improved stability and quality of GAN-generated data, as well as expanding the scope of GAN applications to areas such as video synthesis and natural language processing [95].

GAN is a specific kind of neural network design that is made up of two networks: a generator network and a discriminator network as seen in Fig. 2. The objective of the generator network is to produce realistic samples that accurately represent the distribution of real data, whereas the discriminator network seeks to differentiate between real and fabricated samples [32]. A random noise vector is used as an input to the generator network, which then produces a sample that is meant to be indistinguishable from genuine samples. The discriminator network receives as input a sample, which may have been created by the generator network or may be a real sample, and attempts to determine whether or not the sample is authentic [81].

Both the generator network and the discriminator network are trained using a dataset consisting of both real and fabricated examples throughout the training phase. In addition, the discriminator network is fed a set of fake samples. After that, the generator network is trained to produce bogus samples that are harder for the discriminator network to differentiate from real ones. In addition, the discriminator network undergoes training to improve its ability to differentiate between authentic and bogus data [63].

There are several different ways in which the architecture of a GAN can be altered, such as by adding more layers, tweaking the activation functions, or modifying the

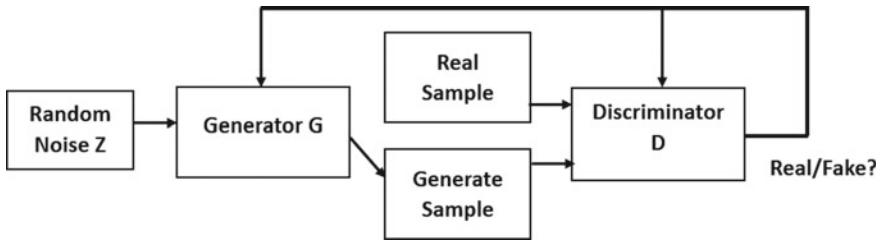


Fig. 2 GAN architecture

loss functions. GANs offer a wide variety of applications in many different fields of study. The many different uses of GANs are outlined in Table 1. The potential for GANs is enormous and continues to expand as researchers investigate new use cases.

BigGANs were the models that Brock et al. [7] introduced. These models realized the effort of creating high-resolution and unique images from diverse datasets. The

Table 1 Application of GANs

No.	Application of GAN	Explanations
1	Image and video generation [75, 89]	New and realistic images and movies can be generated with the use of GANs, such as fake human faces, landscapes, and animals
2	Data augmentation [14, 72]	When real data is insufficient, GANs can be used to generate synthetic data for training machine learning models. This can increase model performance
3	Style transfer [54, 96]	Converting a snapshot into a painting or a sketch is only one example of how GANs can be used
4	Anomaly detection [31, 84]	GANs can be used to detect anomalies in data, such as identifying fraudulent transactions or defective products
5	Virtual reality and gaming [74, 83]	In the realm of virtual reality and video games, GANs can be utilized to build convincing settings and characters
6	Medical image analysis [39, 42, 93]	To better train machine learning models for disease and abnormality detection, GANs can produce synthetic medical images
7	Text-to-image synthesis [57, 88, 91]	GANs can produce visuals based on textual descriptions; for example, they can be used to create a visual depiction of a tale
8	Fashion and product design [35, 86, 87]	GANs have the potential to be utilized to generate new designs for products and fashion, such as the creation of one-of-a-kind apparel designs or the generation of new automotive designs

image recognition algorithm can handle photos with a high resolution as well as a diverse variety of examples taken from the challenging dataset known as ImageNet.

Karras et al. [36] proposed an alternate generator architecture they named StyleGAN. Using the most recent data from all convolutional layers, scientists created a new generator architecture that dynamically adjusts the aesthetic of the output image. Starting at a low resolution and gradually increasing to a high one helps direct the full-picture synthesis process.

Jetchev et al. [6] presented a new architecture for generating textures called a spatial GAN (SGAN). This method can produce high-quality texture images by merging many different source images to produce intricate textures. Two key benefits distinguish Least Squares Generative Adversarial Networks (LSGANs) from more conventional Generative Adversarial Networks. LSGANs have the potential to produce images of higher quality than ordinary GANs, which is the first advantage of using these models. The first advantage of using an LSGAN is that it is possible to produce images of a higher quality than those produced by a standard GAN. Second, LSGANs are more stable in their performance throughout the learning process [10, 59]. Training GANs is a difficult challenge due to the instability of GAN learning. Recently, many articles have shown that the goal function affects the uncertainty of GANs learning [56]. Reducing the conventional GAN objective function can cause gradient loss concerns, which makes it hard to update the generator. To go around this restriction, LSGANs impose a penalty on samples that are too close to the decision border. This causes the generator to produce more gradients whenever it is updated. Furthermore, theoretically establish that the training instability of classic GANs is associated with the objective function's propensity to mode-seek, and that LSGANs exhibit less mode-seek activity. Even though LSGANs exhibit weaker mode-seeking behavior, this step is still recommended. Table 2 describes the advantages and disadvantages of GANs model including the StyleGAN, LSGAN, BigGAN, WGAN, DCGAN, Vanilla GAN, and Conditional GAN.

3 Formula of Various GANs Model

In these sections we will explore the formula for each GAN models. Table 3 describes the formula of each GAN in detail. StyleGAN is a refinement of progressive GAN, the underlying architecture that enables the generation of exceptionally sharp and detailed images. An MLP network is used to learn image styles, and noise is injected at each layer to generate stochastic variations, as stated in [paper]. This is the sole change to the generator design in StyleGAN [36]. Both the generation of high-resolution images through the use of Progressive Growing and the incorporation of image styles into each layer through the use of AdaIN are qualities that the StyleGAN network possesses.

BigGAN is a special kind of generative adversarial network made specifically for producing large numbers of high-quality photos at high resolution. There are several minor tweaks and new features included. The innovations are as follows: (1)

Table 2 Advantages and disadvantages of GANs model

No.	GAN model	Advantages	Disadvantages
1	StyleGAN [71, 62, 66, 80]	<ul style="list-style-type: none"> High-quality image generation: StyleGAN can generate high-quality images with fine details, such as realistic human faces and animals Diverse image generation: StyleGAN can generate diverse images, meaning that it can create multiple variations of a single image, which is useful in applications such as art and fashion Fine-grained control: StyleGAN allows for fine-grained control over the generated images, such as controlling the pose and facial expression of generated faces Transfer learning: StyleGAN can be used as a pre-trained model for image generation tasks, which can significantly reduce the training time for new models 	<ul style="list-style-type: none"> To train and produce images of excellent quality, StyleGAN necessitates many computational resources, such as a GPU Large memory requirement: StyleGAN requires a large amount of memory to store the trained models, which can be challenging for devices with limited memory Limited dataset size: StyleGAN requires a large and diverse dataset for training to avoid overfitting, which can be a challenge in some domains The images that are created by StyleGAN are not necessarily interpretable or explainable, which might be a difficulty for applications that place a premium on interpretability

(continued)

Table 2 (continued)

No.	GAN model	Advantages	Disadvantages
2	LSGAN [1, 16, 43]	<ul style="list-style-type: none"> More stable training: LSGAN has been shown to provide more stable training compared to traditional GANs, which can be prone to mode collapse and other issues Improved image quality: LSGAN has been shown to generate higher quality images compared to traditional GANs, with fewer artifacts and more realistic textures Robust to hyperparameter choices: LSGAN is relatively insensitive to hyperparameter choices, making it easier to train and fine-tune for different datasets Improved convergence: LSGAN has been shown to converge faster and more reliably than traditional GANs, reducing the overall training time 	<ul style="list-style-type: none"> Limitations in diversity: LSGAN may struggle to generate diverse images, which can limit its use in certain applications where diversity is essential Requires more computational resources: LSGAN can be computationally intensive and require more computational resources than traditional GANs Like other GANs, LSGAN's generated images may not be clearly interpretable, which can restrict its usefulness in some contexts Sensitive to initialization: LSGAN can be sensitive to initialization and requires careful tuning of the network architecture and hyperparameters

(continued)

Table 2 (continued)

No.	GAN model	Advantages	Disadvantages
3	BigGAN [22, 34, 58, 60]	<ul style="list-style-type: none"> High-resolution image generation: BigGAN can generate high-resolution images up to 512 × 512 pixels, which is useful in applications such as art, fashion, and advertising Diverse image generation: BigGAN can generate diverse images with fine details, meaning that it can create multiple variations of a single image, which is useful in applications such as art and fashion In comparison to previous GANs, the images generated by BigGAN are of greater quality, with fewer artifacts and more realistic texturing Fine-grained control: BigGAN allows for fine-grained control over the generated images, such as controlling the size and shape of objects in the generated images 	<ul style="list-style-type: none"> Computationally intensive: BigGAN requires a significant number of computational resources, such as GPU, to train and generate high-quality images Large memory requirement: BigGAN requires a large amount of memory to store the trained models, which can be challenging for devices with limited memory BigGAN's generated graphics aren't always explainable or interpretable, which can be a problem for applications where that's a need Dataset size: BigGAN requires a large and diverse dataset for training to avoid overfitting, which can be a challenge in some domains

(continued)

Table 2 (continued)

No.	GAN model	Advantages	Disadvantages
4	WGAN [17, 28, 40, 42]	<ul style="list-style-type: none"> Improved stability: WGAN is more stable than traditional GANs because it uses a different loss function that allows for more effective gradient updates and avoids vanishing gradients Improved convergence: WGAN has been shown to converge faster and more reliably than traditional GANs, reducing the overall training time Better image quality: WGAN has been shown to generate higher quality images compared to traditional GANs, with fewer artifacts and more realistic textures Fine-grained control: WGAN allows for fine-grained control over the generated images, such as controlling the size and shape of objects in the generated images 	<ul style="list-style-type: none"> Requires a substantial quantity of computer resources, such as a GPU, to train and produce high-quality images. WGAN is a computationally expensive algorithm Sensitive to hyperparameters: WGAN is sensitive to hyperparameters, and tuning the hyperparameters can be challenging Limited interpretability: The generated images from WGAN are not always interpretable or explainable, which can be a challenge for applications where interpretability is important Diversity issues: WGAN may have difficulty generating diverse images, limiting its usefulness in contexts where diversity is crucial

(continued)

Table 2 (continued)

No.	GAN model	Advantages	Disadvantages
5	DCGAN [17, 18, 21, 41]	<ul style="list-style-type: none"> To substitute pooling layers, DCGAN uses stride convolutions on the discriminator and fractional convolutions on the generator. In most cases, CNN is utilized to extract the features In order to find a solution to the gradient issues Batch Standardization Algorithm is what DCGAN utilizes to process data. The BN technique corrects weak initializations, brings the gradient to each layer, and prevents the generator from collecting all samples to the equivalent level. All these improvements take place at once. DCGAN uses various activation functions, including Adam optimization, ReLU activation function, and leakyReLU The findings support the efficacy of the GAN structure in sample generation and demonstrate the superior performance of DCGAN. When comparing various GAN models, DCGAN is typically considered to be the gold standard [24] 	<ul style="list-style-type: none"> Computing resources: DCGAN training and high-quality image generation necessitate substantial computing resources, such as GPU DCGAN is hyperparameter sensitive, hence it can be difficult to tune the hyperparameters The images produced by DCGAN are not necessarily interpretable or explainable, which might be problematic for applications where interpretability is crucial It can be difficult to train a deep convolutional generative adversarial network (DCGAN) in some fields due to the need for a large and diverse dataset

(continued)

Table 2 (continued)

No.	GAN model	Advantages	Disadvantages
6	Vanilla GAN [3, 9, 23, 27]	<ul style="list-style-type: none"> • Produces high-quality synthetic data: Vanilla GAN can generate high-quality synthetic data that is visually like real data • Flexible and versatile: GANs can be used for various applications such as image and text generation, image-to-image translation, and even game AI • Unsupervised learning: Unlike supervised learning, GANs do not require labeled data and can generate synthetic data without the need for a training dataset 	<ul style="list-style-type: none"> • Unstable training process: The training process of GANs can be unstable, and it may be difficult to find the optimal hyperparameters and architecture • Mode collapse: GANs can suffer from mode collapse, where the generator only learns to generate a limited set of data samples, rather than the full data distribution • Difficulty in evaluation: It can be difficult to evaluate the quality of generated data, as there is no clear metric for measuring how closely the generated data resembles real data
7	Conditional GAN [13, 15, 49, 64]	<ul style="list-style-type: none"> • Better control over the generated data: cGANs can generate data that is specific to a particular condition, which allows for better control over the generated data • Can generate more diverse data: By conditioning the generator on additional information, cGANs can produce more diverse and realistic data than traditional GANs • Useful for various applications: cGANs can be used for a variety of applications, such as image-to-image translation, text-to-image synthesis, and style transfer 	<ul style="list-style-type: none"> • Requires additional data: cGANs require additional data, such as class labels or text descriptions, to condition the generator, which can be difficult to obtain in some cases • More complex training: The training process for cGANs is more complex than traditional GANs, as the generator must be conditioned on additional information • Overfitting: If the conditioning information is overfit to the training data, the generator may not be able to generate diverse and realistic data

Table 3 Formula of GANs model

No.	GAN model	Formula	Parameter
1	StyleGAN	<ul style="list-style-type: none"> • $AdaIN(x_i, y) = y_{s,i} \frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i}$ (2) 	<ul style="list-style-type: none"> The following formula in StyleGAN makes use of the variable AdaIN. The idea of applying a linear transformation determined by style to the information about the normalized content has not been modified. However, instead of the standard deviation and the average value of the style, y_s and y_b, which are linear transformations to the style vector W and will be described further on, will be used
2	LSGAN	<ul style="list-style-type: none"> $\min_D VLSGAN(D) = \frac{1}{2} E \left[(D(X_{real,i}) - 1)^2 \right]$ (3) $+ \frac{1}{2} E \left[(D(G(X_{fake,i}))^2 \right]$ $\min_G VLSGAN(G) = \frac{1}{2} D \left[(G(X_{fake,i}) - 1)^2 \right]$ (4) 	<ul style="list-style-type: none"> Where E is the expectation, z is the input noise vector, G is the generator network, D is the discriminator network, x is a real image, and 1 and 0 are the target scores for real and fake images, respectively
3	BigGAN	<ul style="list-style-type: none"> $R_\beta(W) = \beta W^T W - I _F^2$ (5) 	<ul style="list-style-type: none"> Where W is a weight matrix and β a hyperparameter

(continued)

Table 3 (continued)

No.	GAN model	Formula	Parameter
4	WGAN	<ul style="list-style-type: none"> $\nabla_w \frac{1}{m} \sum_{i=1}^m [f(x^{(i)}) - f(G(z^{(i)}))]$ (6) $\nabla_\theta \frac{1}{m} \sum_{i=1}^m f(G(z^{(i)}))$ (7) 	<ul style="list-style-type: none"> The 1-Lipschitz function is denoted by the letter f. The constraint is enforced by WGAN by applying a straightforward clipping operation, which lowers the greatest weight value in f. The hyperparameters c are responsible for ensuring that the weights of the discriminator remain within a predetermined range. Further, z represents random noise, G represents generator, $G(z)$ represents samples generated by the generator, C represents discriminator, C^* represents an approximate expression of Wasserstein-1 distance
5	DCGAN	<ul style="list-style-type: none"> The training of DCGAN is expressed in formula (1) as follows (I. J. [30]): $\min i \max(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$ (8) Consequently, when optimizing D, it is to maximum $V(D, G)$, and when optimizing G, it is to minimum $V(D, G)$. Lastly, the optimization problem is displayed in formula (9) and formula (10): $D_G^* = \arg \max V(G, D)$ (9) $G^* = \arg \min V(G, D_G^*)$ (10) 	<ul style="list-style-type: none"> X denotes the first image, z is a d-dimensional vector consisting of arbitrary numbers, and $p_{\text{data}}(x)$ and $p_z(z)$ are the probability distributions of x and z. The probability of the input being a generated image from $p_{\text{data}}(x)$ is $D(x)$, and $(1 - D(G(z)))$ is the probability of being generated from $p_z(z)$ D is trained to have a higher percentage of right answers, while G is trained to have a lower $\log(1 - D(G(z))$ in order to trick D
6	Vanilla GAN	<ul style="list-style-type: none"> $V(G, D) = \frac{1}{x} \int P_{\text{data}}(x) \log(D(x)) dx + pg(x) \log(1 - D(x)) dx$ (11) 	<ul style="list-style-type: none"> E is the expectation, z is the input noise vector, G is the generator network, D is the discriminator network, x is a real image, and 1 and 0 are the target scores for real and fake images, respectively
7	Conditional GAN	<ul style="list-style-type: none"> $\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x y)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z y)))]$ (12) 	<ul style="list-style-type: none"> The expectation represents with E, z is the input noise vector, G is the generator network, D is the discriminator network, x is a real image, and 1 and 0 are the target scores for real and fake images, respectively

Increasing batch sizes, which has a big effect on the Inception Score of the model. (2) Increasing the width in each layer leads to a further Inception Score improvement. (3) Adding skip connections from the latent variable to further layers helps performance. (4) A new variant of Orthogonal Regularization [7].

LSGAN stands for Least Squares Generative Adversarial Networks. It is a type of GAN (Generative Adversarial Network) which was introduced by Mao et al. in their 2017 paper titled “Least Squares Generative Adversarial Networks”. The main idea behind LSGAN is to use a different loss function for the discriminator in the GAN training process. Instead of the binary cross-entropy loss used in traditional GANs, LSGAN uses the least squares loss, which is a type of regression loss. The authors of the LSGAN paper argue that using the least squares loss for the discriminator leads to more stable training and better results in terms of generated image quality. This is because the least squares loss is less sensitive to outliers than the binary cross-entropy loss, which can be helpful in cases where the discriminator is too confident in its classification of real and fake images. Overall, LSGAN has been shown to be an effective approach for training GANs, and has been used in several applications, including image generation and video synthesis.

Vanilla GAN, also known as original GAN, is a type of GAN introduce by Ian Goodfellow et al. in their 2014 paper. The main idea behind Vanilla GAN is to train two neural networks, a generator, and a discriminator, to compete against each other in a two-player minimax game. The generator tries to generate fake samples that resemble the real data, while the discriminator tries to distinguish between real and fake samples. During training, the generator receives random noise as input and generates fake samples, which are then fed into the discriminator along with real samples from the dataset. The discriminator then assigns a probability score to each sample, indicating whether it is real or fake. The generator is then updated based on the feedback from the discriminator to produce more realistic samples. Vanilla GAN uses binary cross-entropy loss function to train the discriminator and generator. The generator tries to maximize the loss, while the discriminator tries to minimize it. However, this method can be unstable and lead to the problem of mode collapse, where the generator learns to produce only a few types of samples. Despite its limitations, Vanilla GAN has been used successfully in a variety of applications, including image generation, style transfer, and data augmentation.

Conditional GAN (cGAN) is a type of GAN that allows for the generation of samples based on specific conditional information. It was introduced by Mehdi Mirza and Simon Osindero in their 2014 paper titled “Conditional Generative Adversarial Nets”. In cGAN, both the generator and discriminator receive additional input information, known as conditional information, which is used to guide the generation of the samples. The conditional information can be any type of auxiliary information, such as labels, class information, or attributes. During training, the generator takes both the random noise and the conditional information as input and generates fake samples. The discriminator receives both the real and fake samples along with the corresponding conditional information and tries to distinguish between them. The generator is updated based on the feedback from the discriminator to produce more realistic samples that also satisfy the conditional information. The loss function

in cGAN is modified to include the conditional information. The generator tries to maximize the log-likelihood of the correct conditional distribution, while the discriminator tries to maximize the log-likelihood of the correct conditional distribution for real samples and the incorrect conditional distribution for fake samples. cGAN has been used successfully in a variety of applications, such as image-to-image translation, text-to-image synthesis, and video prediction. It has also been extended to semi-supervised learning, where only a small portion of the training data is labelled.

4 Review of the Applications of GANs to Synthetic Image Generation Studies

GANs have shown promising results in synthesizing high-quality images. Table 4 presents an overview of the application of GANs in image synthesis. The generation of realistic images has been a longstanding goal in computer vision and machine learning. However, traditional image generation methods such as autoregressive models and variational autoencoders suffer from limited scalability and often produce blurry or unrealistic images. GANs are a type of deep learning algorithm that can generate high-quality images by training a generator network to produce realistic images that can fool a discriminator network.

Ferreira et al. [26] are building a unique style-based deep GAN model called PetroGAN. The goal of this model is to create the first realistic synthetic petrographic datasets across a variety of rock types. To allow for robust replication of statistical and esthetical properties and to improve the internal variance of petrographic data, PetroGAN incorporates the architecture of StyleGAN2 with adaptive discriminator augmentation (ADA). This combination of features is known as adaptive discriminator augmentation. Their research reveals that GANs are an effective strategy for providing synthetic data that is realistic in the field of geosciences. This conclusion can be drawn from the fact that GANs have been shown to be effective. Using a GAN, [2] were able to generate knee X-ray images that accurately reflected the characteristics of the arthritis progression stage, which neither human experts nor artificial intelligence could discern apart from the real images. In a nutshell, the results of their study suggest that it would be possible to implement a generative model to generate anonymous, realistic images that, in addition to resolving data shortages, may also address class disparities. The author [96] proposes a novel segmentation guided style-based GAN for PET synthesis. To increase the performance of the translation, the author uses a task-driven technique that couples a segmentation job with a GAN framework. Extensive experiments demonstrate that our overall framework is superior in PET synthesis, particularly on those regions of interest.

An improved LSGAN model is proposed in [89]. This model generates GPR pictures by combining the loss functions of LSGAN with convolutional neural networks (CNN). This model has the capability of generating high-precision GPR data, which can be used to address the dearth of labelled GPR data. The author

Table 4 Review of the applications of GANs to synthetic image generation studies

References	Model	Applications
[26]	StyleGAN	Synthetic petrographic datasets
[2]	StyleGAN	High-resolution knee plain radiography image synthesis
[96]	StyleGAN	Positron emission tomography (PET) synthesis
[89]	LSGAN	Generation of high-precision ground penetrating radar images
[46]	LSGAN	A study on bearing fault diagnosis
[58]	BigGAN	Reconstruction of natural images from human brain activity
[85]	BigGAN	Deep generative representations for scene synthesis
[77]	BigGAN	Fake face detection
[33]	DCGAN	High-content image generation for drug discovery
[3]	DCGAN	Brain MRI image generation
[17]	DCGAN	Traffic sign synthetic data generation
[28]	WGAN	Training of dataset in industrial process
[78]	WGAN	Lung nodules in CT images
[50]	WGAN	Perceptual image restoration
[55]	Vanilla GAN	Generating images for supervised hyperspectral image classification
[68]	Vanilla GAN	Lung cancer diagnosis using Hessian adaptive learning optimization
[37]	Vanilla GAN	Synthetic attack data generation
[49]	Conditional GAN	Data augmentation for audio–visual emotion recognition
[64]	Conditional GAN	Large mask image completion
[38]	Conditional GAN	Multimodal neuroimaging synthesis
[92]	CycleGAN	Multi-contrast MRI image synthesis
[11]	CycleGAN	A surface defect detection system
[75]	CycleGAN	Improving de novo molecule generation
[70]	InfoGAN	Synthetic CT image generation of shape-controlled lung cancer
[61]	InfoGAN	Image retrieval
[25]	InfoGAN	Synthetic aperture radar (SAR) image

conducts an analysis of the suggested model by employing the Frechet Inception Distance (FID) assessment index. He then compares the proposed model to other existing GAN models and discovers that it outperforms the other two models despite having a lower FID score.

A failure detection approach for rolling bearings is proposed in [46], and it makes use of conditional deep convolution adversarial generative networks (C-DCGAN)

for effective data augmentation. The results of the experiment suggest that the diagnostic approach presented in this research has the potential to enhance the fault classification impact of rolling bearings by creating sample data that is balanced and sufficient. This could be done because of the findings of the experiment which show that the diagnostic method has the potential to improve the fault classification impact of rolling bearings. Further, [58] present a novel GAN-based Bayesian visual reconstruction model (GAN-BVRM) in order to circumvent the inherent tension that exists between naturalness and fidelity in the GAN-based methods that are already in use. The components of GAN-BVRM include a classifier to decode the categories from the fMRI data, a pre-trained conditional generator of the distinguishing BigGAN to generate natural images of the specified categories, an evaluator to assess the quality of the generated images, and an evaluator to assess the quality of the generated text. To fit the voxels of the fMRI scan, GAN-BVRM iteratively updates the noise input vector using backpropagation, making it fully differentiable in the process. GAN-BVRM is also totally differentiable due to its neural network construction.

The author [3] proposed a system for generating and categorizing brain MRI images using GAN architectures and deep learning models was suggested. This framework is called BrainGAN, and that's what it's called in the study. Because of this finding, the researchers came up with a way for images to be instantly checked to see if they meet the standards. In this method, CNN, MobileNetV2, and ResNet152V2 are the three models that are used. The deep transfer models were trained on images made by Vanilla GAN and DCGAN. Their success was then measured on a test set made up of real brain MRI scans. Based on the results of the testing, it was found that the ResNet152V2 model did much better than the other two models. Based on the brain MRI images made by the DCGAN design, the ResNet152V2 had an accuracy of 99.09%, precision of 99.12%, recall of 99.08%, area under the curve (AUC) of 99.51%, and a loss of 0.196.

Ma et al. [50] try to enhance the WGAN training by imposing a pairwise restriction on the critic. This constraint is geared for picture restoration challenges. Whenever there is a choice to be made between two images, the pairwise constraint encourages the critic to give a higher rating to the original (actual) image as opposed to the restored (produced) image. This is the case anytime there is a decision to be made between two different pictures. We show that such a pairwise constraint can be applied by reversing the gradients in the WGAN training, which leads to the proposed corrected Wasserstein generative adversarial network (ReWaGAN).

An approach known as Hessian Adaptive Learning, or HAL Optimization, was suggested by [68]. The HL technique that has been proposed makes use of gradient and curvature data to solve the problem of mode collapse and to increase the quantity of the dataset by generating a variety of images. Vanilla GAN, CGAN, WGAN, and DCGAN were used in the tests. To test each GAN, we apply the stochastic gradient descent (SGD), the Gauss–Newton (GN) second-order learning, and the recommended HAL optimization techniques. The findings of the trials show that the GAN models optimized with the HAL technique perform better than the GN and SGD models in terms of overall performance. This is demonstrated by the fact that the GAN models are superior. The findings of the trials showed that GANs

swiftly converge, and they also highlighted that difficulties with mode collapse may be overcome by performing HAL optimization.

Ma et al. [49] construct a multimodal conditional GAN for audio-visual emotion recognition and present an effective method for data augmentation based on this network. They oversee the creation of modality generators and discriminators for both auditory and visual information. The information about the category is used as the shared input for our GAN, which helps to ensure that it can produce fake data for a wide variety of categories. The high dependence that exists between the auditory modality and the visual modality in the multimodal data that was obtained is modelled using the Hirschfeld-Gebelein-Renyi (HGR) maximal correlation. This model was used to model the data that was collected. In this method, they can approximate the genuine data by relating the many modalities that are present in the created data. After that, the generated data are added to our data manifold to make it more complete. We further apply our strategy to address the issue of class disparity. This is the first piece of research that we are aware of that suggests a data augmentation strategy for audio-visual emotion identification utilizing a multimodal conditional GAN.

5 Conclusions

In this paper, the author conducts an in-depth analysis and evaluation of a variety of GAN models and architectural frameworks utilized within the process of synthetic image production. According to the authors' knowledge, the current publication is the first study within this research topic that summarizes and categorizes the existing literature for the purpose of providing a more comprehensive overview. GAN is a powerful method for generating synthetic images. In this review, we have discussed several variants of GANs, including Vanilla GAN, Conditional GAN (cGAN), WGAN, BigGAN, DCGAN, LSGAN, and StyleGAN. Vanilla GAN uses binary cross-entropy loss function to train the discriminator and generator, which can be unstable and lead to the problem of mode collapse. Further, cGAN addresses this issue by allowing the generation of samples based on specific conditional information. WGAN uses the Wasserstein distance to improve training stability and reduce the problem of mode collapse. GANs have been used successfully in various applications, such as image synthesis, image editing, and data augmentation. They have also been extended to other domains, such as text, speech, and music.

However, GANs still have some limitations and challenges. The training process can be time-consuming and require a large amount of data. The generated samples may also suffer from issues such as blurriness and lack of diversity. Additionally, GANs can be sensitive to hyperparameters and require careful tuning. Overall, GANs hold great potential for synthetic image generation and have already shown impressive results in various applications. As GANs continue to evolve, we can expect even more advanced and sophisticated image generation techniques to emerge in the future.

References

1. Abu-Srhan, A., Abushariah, M. A. M., & Al-Kadi, O. S. (2022). The effect of loss function on conditional generative adversarial networks. *Journal of King Saud University—Computer and Information Sciences* 34(9). <https://doi.org/10.1016/j.jksuci.2022.02.018>.
2. Ahn, G., Choi, B. S., Ko, S., Jo, C., Han, H. S., Lee, M. C., & Ro, D. H. (2023). High-resolution knee plain radiography image synthesis using style generative adversarial network adaptive discriminator augmentation. *Journal of Orthopaedic Research*, 41(1), 84–93. <https://doi.org/10.1002/jor.25325>
3. Alrashedy, H. H. N., Almansour, A. F., Ibrahim, D. M., & Hammoudeh, M. A. A. (2022). BrainGAN: Brain MRI image generation and classification framework using GAN architectures and CNN models. *Sensors* 22(11). <https://doi.org/10.3390/s22114297>.
4. Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. In *34th international conference on machine learning, ICML 2017*, pp. 298–321
5. Bau, D., Zhu, J. Y., Strobelt, H., Zhou, B., Tenenbaum, J. B., Freeman, W. T., & Torralba, A. (2019). GaN dissection: Visualizing and understanding generative adversarial networks. In *7th international conference on learning representations, ICLR 2019*, pp. 1–18.
6. Bergmann, U., Jetchev, N., & Vollgraf, R. (2017). Learning texture manifolds with the periodic spatial GaN. *ArXiv* 1.
7. Brock, A., Donahue, J., & Simonyan, K. (2019). Large scale Gan training for high fidelity natural image synthesis. In *7th international conference on learning representations, ICLR 2019*.
8. Cao, C., Cao, Z., & Cui, Z. (2020). LDGAN: A synthetic aperture radar image generation method for automatic target recognition. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5), 3495–3508. <https://doi.org/10.1109/TGRS.2019.2957453>
9. Castelli, M., Manzoni, L., Espindola, T., Popović, A., & Lorenzo, A. D. (2021). Generative adversarial networks for generating synthetic features for Wi-Fi signal quality. *PLoS ONE* 16(11 November). <https://doi.org/10.1371/journal.pone.0260308>.
10. Chen, R.-C., Dewi, C., Zhang, W.-W., & Liu, J.-M. (2020). Integrating gesture control board and image recognition for gesture recognition based on deep learning. *International Journal of Applied Science and Engineering (IJASE)*, 17(3), 237–248.
11. Chen, S. H., Lai, Y. W., Kuo, C.L., Lo, C. Y., Lin, Y. S., Lin, Y. R., Kang, C. H., & Tsai, C. C. (2022). A surface defect detection system for golden diamond pineapple based on cycleGAN and YOLOv4. *Journal of King Saud University—Computer and Information Sciences* 34(10). <https://doi.org/10.1016/j.jksuci.2022.07.018>.
12. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). InfoGAN. *Advances in Neural Information Processing Systems*.
13. Cho, J., Yoon, K. (2020). Conditional activation GAN: Improved auxiliary classifier GAN. *IEEE Access* 8. <https://doi.org/10.1109/ACCESS.2020.3041480>.
14. Chowdhury, S. R., Tornberg, L., Halvfordsson, R., Nordh, J., Gustafsson, A. S., Wall, J., Westerberg, M., Wirehed, A., Tilloy, L., Hu, Z., Tan, H., Pan, M., & Sjoberg, J. (2019). Automated augmentation with reinforcement learning and GANs for robust identification of traffic signs using front camera images. In *Conference Record—Asilomar Conference on Signals, Systems and Computers*.
15. Chrysos, G. G., Kossaifi, J., & Zafeiriou, S. (2020). RoCGAN: Robust conditional GAN. *International Journal of Computer Vision* 128(10–11). <https://doi.org/10.1007/s11263-020-01348-5>.
16. Deng, F., Wan, Q., Zeng, Y., Shi, Y., Wu, H., Wu, Y., Xu, W., Mok, G. S. P., Zhang, X., & Hu, Z. (2022). Image restoration of motion artifacts in cardiac arteries and vessels based on a generative adversarial network. *Quantitative Imaging in Medicine and Surgery* 12(5). <https://doi.org/10.21037/qims-20-1400>.
17. Dewi, C., Chen, R.-C., Liu, Y.-T., & Tai, S.-K. (2021). Synthetic data generation using DCGAN for improved traffic sign recognition. *Neural Computing and Applications*, 33(8), 1–15.

18. Dewi, C., Chen, R.-C., & Liu, Y. T. (2022). Synthetic traffic sign image generation applying generative adversarial networks. *Vietnam Journal of Computer Science*, 9(1), 1–12. <https://doi.org/10.1142/S219688822500191>
19. Dewi, C., Chen, R. C., & Liu, Y. T. (2021). Wasserstein generative adversarial networks for realistic traffic sign image generation. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (vol. 12672, pp. 479–93). LNAI, Springer.
20. Dewi, C., Chen, R. C., Liu, Y. T., Jiang, X., & Hartomo, K. D. (2021). Yolo V4 for advanced traffic sign recognition with synthetic training data generated by various GAN. *IEEE Access*, 9, 97228–97242. <https://doi.org/10.1109/ACCESS.2021.3094201>
21. Ding, H., Cui, Z., Maghami, E., Chen, Y., Matinlinna, J. P., Pow, E. H. N., Fok, A. S. L., Burrow, M. F., Wang, W., Tsai, J. K. H. (2023). Morphology and mechanical performance of dental crown designed by 3D-DCGAN. *Dental Materials* 39(3). <https://doi.org/10.1016/j.dental.2023.02.001>.
22. Dixe, S., Leite, J., Fonseca, J. C., & Borges, J. (2022). BigGAN evaluation for the generation of vehicle interior images. *Procedia Computer Science* 204
23. Duy, P. T., Tien, L. K., Khoa, N. H., Hien, D. T. T., Nguyen, A. G. T., & Pham, V. H. (2021). DIGFuPAS: Deceive IDS with GAN and function-preserving on adversarial samples in SDN-enabled networks. *Computers and Security* 109. <https://doi.org/10.1016/j.cose.2021.102367>.
24. Fang, W., Ding, Y., Zhang, F., & Sheng, J. (2019). Gesture recognition based on CNN and DCGAN for calculation and text output. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2019.2901930>.
25. Feng, Z., Daković, M., Ji, H., Zhou, X., Zhu, M., Cui, X., & Stanković, L. (2023). Interpretation of latent codes in InfoGAN with SAR images. *Remote Sensing* 15(5). <https://doi.org/10.3390/rs15051254>.
26. Ferreira, I., Ochoa, L., & Koeshidayatullah, A. (2022). On the generation of realistic synthetic petrographic datasets using a style-based GAN. *Scientific Reports* 12(1). <https://doi.org/10.1038/s41598-022-16034-4>.
27. Gan, H., Shen, M., Hua, Y., Ma, C., & Zhang, T. (2023). From patch to pixel: A transformer-based hierarchical framework for compressive image sensing. *IEEE Transactions on Computational Imaging* 9. <https://doi.org/10.1109/TCI.2023.3244396>.
28. Gao, S., Qiu, S., Ma, Z., Tian, R., Liu, Y. (2022). SVAE-WGAN-based soft sensor data supplement method for process industry. *IEEE Sensors Journal* 22(1). <https://doi.org/10.1109/JSEN.2021.3128562>.
29. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680.
30. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Bing, X., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets (NIPS version). *Advances in Neural Information Processing Systems*, 27(27), 2672–2680. <https://doi.org/10.1001/jamaintermmed.2016.8245>
31. Han, C., Rundo, L., Murao, K., Noguchi, T., Shimahara, Y., Milacski, Z. Á., Koshino, S., Sala, E., Nakayama, H., & Satoh, S. (2021). MADGAN: Unsupervised medical anomaly detection GAN using multiple adjacent brain MRI slice reconstruction. *BMC Bioinformatics* 22. <https://doi.org/10.1186/s12859-020-03936-1>.
32. Huang, G., & Jafari, A. H. (2023). Enhanced balancing GAN: Minority-class image generation. *Neural Computing and Applications*, 35(7), 5145–5154. <https://doi.org/10.1007/s00521-021-06163-8>
33. Hussain, S., Anees, A., Das, A., Nguyen, B. P., Marzuki, M., Lin, S., Wright, G., Singhal, A. (2020). high-content image generation for drug discovery using generative adversarial networks. *Neural Networks* 132. <https://doi.org/10.1016/j.neunet.2020.09.007>.
34. Jabbar, A., Li, X., Assam, M., Khan, J. A., Obayya, M., Alkhonaini, M. A., Al-Wesabi, F. N., & Assad, M. (2022). AFD-StackGAN: Automatic mask generation network for face de-occlusion using StackGAN. *Sensors* 22(5). <https://doi.org/10.3390/s22051747>.

35. Jo, J., Lee, S., Lee, C., Lee, D., & Lim, H. (2020). Development of fashion product retrieval and recommendations model based on deep learning. *Electronics (Switzerland)* 9(3). <https://doi.org/10.3390/electronics9030508>.
36. Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition* (Vol. 2019–June).
37. Kumar, V., & Sinha, D. (2023). Synthetic attack data generation model applying generative adversarial network for intrusion detection. *Computers and Security* 125. <https://doi.org/10.1016/j.cose.2022.103054>.
38. Lan, H., Toga, A. W., & Sepehrband, F. (2021). Three-dimensional self-attention conditional GAN with spectral normalization for multimodal neuroimaging synthesis. *Magnetic Resonance in Medicine* 86(3). <https://doi.org/10.1002/mrm.28819>.
39. Lan, L., You, L., Zhang, Z., Fan, Z., Zhao, W., Zeng, N., Chen, Y., Zhou, X. (2020). Generative adversarial networks and its applications in biomedical informatics. *Frontiers in Public Health* 8.
40. Lee, S., Kim, J., Lee, G., Hong, J., Bae, J. H., & Lim, K. J. (2021). Prediction of aquatic ecosystem health indices through machine learning models using the wgan-based data augmentation method. *Sustainability (Switzerland)* 13(18). <https://doi.org/10.3390/su131810435>.
41. Li, D., Ling, H., Kim, S. W., Kreis, K., Fidler, S., & Torralba, A. (2022). BigDatasetGAN: Synthesizing ImageNet with pixel-wise annotations. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition* (Vol. 2022–June).
42. Li, X., Jiang, Y., Rodriguez-Andina, J. J., Luo, H., Yin, S., & Kaynak, O. (2021). When medical images meet generative adversarial network: Recent development and research opportunities. *Discover Artificial Intelligence* 1(1). <https://doi.org/10.1007/s44163-021-00006-0>.
43. Li, Y., Zhang, M., & Chen, C. (2022) A deep-learning intelligent system incorporating data augmentation for short-term voltage stability assessment of power systems. *Applied Energy* 308. <https://doi.org/10.1016/j.apenergy.2021.118347>.
44. Liu, B., Lv, J., Fan, X., Luo, J., & Zou, T. (2022). Application of an improved DCGAN for image generation. *Mobile Information Systems* 2022. <https://doi.org/10.1155/2022/9005552>.
45. Liu, F., Xu, M., Li, G., Pei, J., Shi, L., & Zhao, R. (2021). Adversarial symmetric GANs: Bridging adversarial samples and adversarial networks. *Neural Networks* 133. <https://doi.org/10.1016/j.neunet.2020.10.016>.
46. Liu, J., Li, C., Su, Y., & Sun, X. (2022). A study on bearing fault diagnosis based on LSGAN-SqueezeNet. *Zhendong Yu Chongji/Journal of Vibration and Shock* 41(12). <https://doi.org/10.13465/j.cnki.jvs.2022.12.036>.
47. Liu, M. Y., Huang, X., Yu, J., Wang, T. C., & Mallya, A. (2021). Generative adversarial networks for image and video synthesis: Algorithms and applications. *Proceedings of the IEEE* 109(5). <https://doi.org/10.1109/JPROC.2021.3049196>.
48. Liu, Yan Ting, Rung Ching Chen, and Christine Dewi. 2021. “Generate Realistic Traffic Sign Image Using Deep Convolutional Generative Adversarial Networks.” in *2021 IEEE Conference on Dependable and Secure Computing, DSC 2021*.
49. Ma, F., Li, Y., Ni, S., Huang, S., & Zhang, L. (2022). Data augmentation for audio-visual emotion recognition with an efficient multimodal conditional GAN. *Applied Sciences (Switzerland)* 12(1). <https://doi.org/10.3390/app12010527>.
50. Ma, H., Liu, D., & Wu, F. (2023). Rectified wasserstein generative adversarial networks for perceptual image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45(3). <https://doi.org/10.1109/TPAMI.2022.3185316>.
51. Mao, Q., Lee, H. Y., Tseng, H. Y., Ma, S., & Yang, M. H. (2019). Mode seeking generative adversarial networks for diverse image synthesis. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition* (Vol. 2019–June, pp. 1429–1437).
52. Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., & Smolley, S. P. (2019). On the effectiveness of least squares generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(12), 2947–2960. <https://doi.org/10.1109/TPAMI.2018.2872043>

53. Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *CoRR*.
54. Mukherjee, D., Saha, P., Kaplun, D., Sinitca, A., & Sarkar, R. (2022). Brain tumor image generation using an aggregation of GAN models with style transfer. *Scientific Reports* 12(1). <https://doi.org/10.1038/s41598-022-12646-y>.
55. Osman, H. A. A., & Azlan, N. Z. (2022). Generating images for supervised hyperspectral image classification with generative adversarial nets. *Journal of Integrated and Advanced Engineering (JIAE)* 2(2). <https://doi.org/10.5166/jiae.v2i2.80>.
56. Qi, G. (Jun 2020). Loss-sensitive generative adversarial networks on lipschitz densities. *International Journal of Computer Vision*. <https://doi.org/10.1007/s11263-019-01265-2>.
57. Qi, Z., Fan, C., Xu, L., Li, X., & Zhan, S. (2021). MRP-GAN: Multi-resolution parallel generative adversarial networks for text-to-image synthesis. *Pattern Recognition Letters* 147. <https://doi.org/10.1016/j.patrec.2021.02.020>.
58. Qiao, K., Chen, J., Wang, L., Zhang, C., Tong, L., & Yan, B. (2020). BigGAN-based bayesian reconstruction of natural images from human brain activity. *Neuroscience* 444. <https://doi.org/10.1016/j.neuroscience.2020.07.040>.
59. Radford, A., Metz, L., & Chintala, S. (2016). Unsupervised representation learning with deep convolutional GANs. *International Conference on Learning Representations*, pp. 1–16. <https://doi.org/10.1101/0004-6361/201527329>.
60. Ravuri, S., & Vinyals, O. (2019). Seeing is not necessarily believing: limitations of BigGANs for data augmentation. *International Conference on Learning Representations (ICLR)*.
61. Sabry, E. S., Elagooz, S. S., El-Samie, F. E. A., El-Shafai, W., El-Bahnasawy, N. A., El-Banby, G. M., Algarni, A. D., Soliman, N. F., Ramadan, R. A. (2023). Image retrieval using convolutional autoencoder, InfoGAN, and vision transformer unsupervised models. *IEEE Access* 11. <https://doi.org/10.1109/ACCESS.2023.3241858>.
62. Hussin, S., Hussin, S., & Yildirim, R. (2021). StyleGAN-LSRO method for person re-identification. *IEEE Access*, 9, 13857–13869. <https://doi.org/10.1109/ACCESS.2021.3051723>
63. Santis, J. A., Marín-García, C. A., & Sánchez-R, V. M. (2023). Effect of different substrates on material properties of cubic GaN thin films grown by LP-MOCVD method. *Journal of Crystal Growth* 601. <https://doi.org/10.1016/j.jcrysrgo.2022.126944>.
64. Shao, C., Li, X., Li, F., & Zhou, Y. (2022). Large mask image completion with conditional GAN. *Symmetry* 14(10). <https://doi.org/10.3390/sym14102148>.
65. Song, J., Zhang, J., Gao, L., Zhao, Z., Shen, H. T. (2022). AgeGAN++: face aging and rejuvenation with dual conditional GANs. *IEEE Transactions on Multimedia* 24. <https://doi.org/10.1109/TMM.2021.3059336>.
66. Su, W., Ye, H., Chen, S. Y., Gao, L., & Fu, H. (2022). DrawingInStyles: portrait image generation and editing with spatially conditioned StyleGAN. *IEEE Transactions on Visualization and Computer Graphics*. <https://doi.org/10.1109/TVCG.2022.3178734>.
67. Sun, D., Yang, K., Shi, Z., & Chen, C. (2018). A new mimicking attack by LSGAN. In *Proceedings—international conference on tools with artificial intelligence, ICTAI*.
68. Thirumagal, E., & Saruladha, K. (2023). Lung cancer diagnosis using hessian adaptive learning optimization in generative adversarial networks. *Soft Computing*. <https://doi.org/10.1007/s00500-023-07877-8>.
69. Tian, Y., Yang, G., Wang, Z., Li, E., & Liang, Z. (2019). Detection of apple lesions in orchards based on deep learning methods of cyclegan and YoloV3-dense. *Journal of Sensors*. <https://doi.org/10.1155/2019/7630926>.
70. Toda, R., Teramoto, A., Tsujimoto, M., Toyama, H., Imaizumi, K., Saito, K., & Fujita, H. (2021). Synthetic CT image generation of shape-controlled lung cancer using semi-conditional InfoGAN and its applicability for type classification. *International Journal of Computer Assisted Radiology and Surgery* 16(2). <https://doi.org/10.1007/s11548-021-02308-1>.
71. Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., & Cohen-Or, D. (2021). Designing an encoder for StyleGAN image manipulation. *ACM Transactions on Graphics* 40(4). doi: <https://doi.org/10.1145/3450626.3459838>.
72. Tran, N. T., Tran, V. H., Nguyen, N. B., Nguyen, T. K., & Cheung, N. M. (2021). On data augmentation for GAN training. *IEEE Transactions on Image Processing* 30. <https://doi.org/10.1109/TIP.2021.3049346>.

73. Volkhonskiy, D., Nazarov, I., & Burnaev, E. (2019). Steganographic generative adversarial networks. In *Proceedings Volume 11433, Twelfth International Conference on Machine Vision (ICMV 2019)* (pp. 1–12).
74. Wafa, A., & Nasipoulous, P. (2022). Light field GAN-based view synthesis using full 4D information. In *Proceedings—CVMP 2022: 19th ACM SIGGRAPH European conference on visual media production*.
75. Wang, F., Feng, X., Guo, X., Xu, L., Xie, L., Chang, S. (2021). Improving de novo molecule generation by embedding LSTM and attention mechanism in CycleGAN. *Frontiers in Genetics* 12. <https://doi.org/10.3389/fgene.2021.709500>.
76. Wang, H., Huang, M., Wu, D., Li, Y., & Zhang, W. (2021). Supervised video-to-video synthesis for single human pose transfer. *IEEE Access* 9. <https://doi.org/10.1109/ACCESS.2021.3053617>.
77. Wang, J., Zeng, K., Ma, B., Luo, X., Yin, Q., Liu, G., & Jha, S. K. (2022). GAN-generated fake face detection via two-stream CNN with PRNU in the wild. *Multimedia Tools and Applications* 81(29). <https://doi.org/10.1007/s11042-021-11592-7>.
78. Wang, Q., Zhou, X., Wang, C., Liu, Z., Huang, J., Zhou, Y., Li, C., Zhuang, H., & Cheng, J. Z. (2019). WGAN-based synthetic minority over-sampling technique: Improving semantic fine-grained classification for lung nodules in CT images. *IEEE Access* 7. <https://doi.org/10.1109/ACCESS.2019.2896409>.
79. Wei, G., Luo, M., Liu, H., Zhang, D., & Zheng, Q. (2020). Progressive generative adversarial networks with reliable sample identification. *Pattern Recognition Letters*, pp. 91–98.
80. Wei, T., Chen, D., Zhou, W., Liao, J., Weiming Zhang, L., Yuan, G. H., & Nenghai, Y. (2022). E2Style: Improve the efficiency and effectiveness of StyleGAN inversion. *IEEE Transactions on Image Processing*, 31, 3267–3280. <https://doi.org/10.1109/TIP.2022.3167305>
81. Won, D. O., Jang, Y. N., & Lee, S. W. (2023). PlausMal-GAN: Plausible malware training based on generative adversarial networks for analogous zero-day malware detection. *IEEE Transactions on Emerging Topics in Computing* 11(1). <https://doi.org/10.1109/TETC.2022.3170544>.
82. Wu, S., Tang, H., Jing, X. Y., Qian, J., Sebe, N., Yan, Y., Zhang, Q. (2022). Cross-view panorama image synthesis with progressive attention GANs. *Pattern Recognition* 131. <https://doi.org/10.1016/j.patcog.2022.108884>.
83. Wu, Y., Zhang, M., Li, X., Gan, Y., & Zhao, C. (2021). Augment reality-based teaching practice. *Biomedical Engineering Education* 1(1). <https://doi.org/10.1007/s43683-020-00040-x>.
84. Xia, X., Pan, X., Li, N., He, X., Ma, L., Zhang, X., & Ding, N. (2022). GAN-based anomaly detection: A review. *Neurocomputing* 493.
85. Yang, C., Shen, Y., & Zhou, B. (2021). Semantic hierarchy emerges in deep generative representations for scene synthesis. *International Journal of Computer Vision* 129(5). <https://doi.org/10.1007/s11263-020-01429-5>.
86. Yang, C., Zhou, Y., Zhu, B., Yu, C., & Wu, L. (2021). Emotionally intelligent fashion design using CNN and GAN. *Computer-Aided Design and Applications* 18(5). <https://doi.org/10.14733/cadaps.2021.900-913>.
87. Yuan, C., & Moghaddam, M. (2020). Attribute-aware generative design with generative adversarial networks. *IEEE Access* 8. <https://doi.org/10.1109/ACCESS.2020.3032280>.
88. Yuan, M., & Peng, Y. (2020). Bridge-GAN: Interpretable representation learning for text-to-image synthesis. *IEEE Transactions on Circuits and Systems for Video Technology* 30(11). <https://doi.org/10.1109/TCSVT.2019.2953753>.
89. Yue, Y., Liu, H., Meng, X., Li, Y., & Du, Y. (2021). Generation of high-precision ground penetrating radar images using improved least square generative adversarial networks. *Remote Sensing* 13(22). <https://doi.org/10.3390/rs13224590>.
90. Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2019). Self-attention generative adversarial networks. In *36th international conference on machine learning, ICML 2019* (pp. 12744–12753).
91. Zhang, H., Zhu, H., Yang, S., & Li, W. (2021). DGattGAN: Cooperative up-sampling based dual generator attentional GAN on text-to-image synthesis. *IEEE Access* 9. <https://doi.org/10.1109/ACCESS.2021.3058674>.

92. Zhang, H., Li, H., Dillman, J. R., Parikh, N. A., & He, L. (2022). Multi-contrast MRI image synthesis using switchable cycle-consistent generative adversarial networks. *Diagnostics* 12(4). <https://doi.org/10.3390/diagnostics12040816>.
93. Zhao, J., Hou, X., Pan, M., & Zhang, H. (2022). Attention-based generative adversarial network in medical imaging: A narrative review. *Computers in Biology and Medicine* 149. <https://doi.org/10.1016/j.combiomed.2022.105948>.
94. Zhou, C., Zhang, J., & Liu, J. (2018). Lp-WGAN: Using Lp-norm normalization to stabilize Wasserstein generative adversarial networks. *Knowledge-Based Systems*, 161, 415–424. <https://doi.org/10.1016/j.knosys.2018.08.004>
95. Zhou, T., Li, Q., Lu, H., Cheng, Q., & Zhang, X. (2023). GAN review: Models and medical image fusion applications. *Information Fusion* 91. <https://doi.org/10.1016/j.inffus.2022.10.017>.
96. Zhou, Y., Yang, Z., Zhang, H., E. I. C. Chang, Fan, Y., & Xu, Y. (2022). 3D segmentation guided style-based generative adversarial networks for PET synthesis. *IEEE Transactions on Medical Imaging* 41(8). <https://doi.org/10.1109/TMI.2022.3156614>.
97. Zhou, Y., Yu, K., Wang, M., Ma, Y., Peng, Y., Chen, Z., Zhu, W., Shi, F., & Chen, X. (2022). Speckle noise reduction for OCT images based on image style transfer and conditional GAN. *IEEE Journal of Biomedical and Health Informatics* 26(1). <https://doi.org/10.1109/JBHI.2021.3074852>

Image Rendering with Generative Adversarial Networks



Fayçal Abbas, Mehdi Malah, and Ramzi Agaba

Abstract This chapter delves into the concepts of neural rendering and generative models, highlighting their importance in the fields of computer graphics, computer vision, and artificial intelligence. Neural rendering techniques utilize deep learning algorithms to generate realistic images, videos, or 3D models, while generative models learn the underlying data distribution to create novel content. The chapter explores various methodologies, such as Neural Radiance Fields, Variational Autoencoders, and Generative Adversarial Networks, among others. Applications of these techniques, such as photorealistic rendering, style transfer, and image synthesis, are discussed, along with the challenges and limitations associated with them. The chapter concludes with an outlook on the future prospects of neural rendering and generative models, emphasizing their potential to revolutionize digital content creation and consumption.

1 Introduction

Neural rendering and generative models are at the forefront of artificial intelligence, playing a significant role in computer graphics, computer vision, and machine learning. These models have the ability to generate new content, such as images, videos, or 3D objects, by learning from existing data. This introduction aims to provide an in-

F. Abbas (✉)
LESI, University of Abbes Laghrour, Khencela, Algeria
e-mail: abbas_faycal@univ-khencela.dz

M. Malah · R. Agaba
ICOSI, University of Abbes Laghrour, Khencela, Algeria
e-mail: malah.mehdi@univ-khencela.dz

R. Agaba
e-mail: ramzi.agaba@univ-oeb.dz

R. Agaba
ReLaCS2 Laboratory, Computer Science Department, University Larbi Ben Mhidi,
Oum El Bouaghi, Algeria

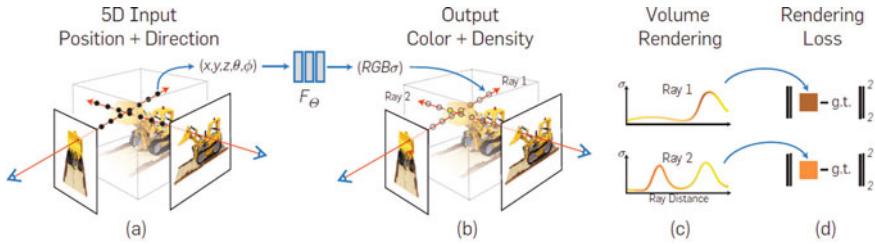


Fig. 1 **a** The first step involves sampling 5D coordinates, which consist of spatial location and viewing direction. **b** A multilayer perceptron (MLP) takes the 5D coordinates as input and outputs a color and volume density for each sampled point. **c** Using volume rendering techniques, the color and volume density values are composited to generate an image. **d** The rendering function used in this procedure is differentiable, allowing for optimization of the scene representation [2]

depth understanding of neural rendering and generative models, their development, and their applications.

Neural rendering refers to a set of techniques that leverage deep learning algorithms to synthesize realistic images, videos, or 3D models [1]. These techniques often involve mapping input data, such as images or geometry, to output images or 3D structures using neural networks. One prominent example of neural rendering is the Neural Radiance Fields (NeRF) approach (see Fig. 1), which learns to represent a 3D scene as a continuous function using a deep neural network [2].

Generative models, on the other hand, are a subset of unsupervised learning methods that aim to learn the underlying data distribution and generate new samples from that distribution [3]. In this chapter, we provide a comprehensive summary of several types of generative models, including Variational Autoencoders (VAEs) [4], Generative Adversarial Networks (GANs) [3], autoregressive models [5].

The development of neural rendering and generative models has been fueled by the rapid advancement of deep learning techniques, particularly Convolutional Neural Networks (CNNs) [6] and Recurrent Neural Networks (RNNs) [7]. These models have demonstrated remarkable performance in various tasks, including image classification, natural language processing, and reinforcement learning.

Recent advancements in neural rendering and generative models have enabled a wide range of applications, such as photorealistic rendering [8], style transfer [9], image synthesis [10], and image-to-image translation [11]. These techniques have been applied to fields as diverse as gaming, animation, and virtual reality, and have the potential to revolutionize the way we create and consume digital content.

2 Neural Rendering

Neural rendering is a technology that uses artificial intelligence (AI) to create realistic images and videos from simulations. It uses a deep learning process, where the AI is trained on a variety of data, such as real-world images, to create high-quality visuals.

The AI then uses the data to generate new images or videos with an impressive level of realism. Neural rendering has become increasingly popular in recent years because it is able to produce photorealistic results in significantly less time than traditional methods. It is also useful for tasks such as object recognition and virtual reality (VR) applications. In addition, neural rendering can be used to generate animations and special effects for movies, television programs, and video games. Neural rendering can be broadly classified into two categories: image-based neural rendering and geometric neural rendering.

2.1 *Image-Based Neural Rendering*

Image generation using Image-based Neural Rendering (INR) is a technique that teaches a system to map input data to output images. INR methods often involve learning a latent space representation and then decoding it to generate the output. The Neural Radiance Fields (NeRF) approach is an example of INR, which learns to represent a 3D scene as a continuous function using a deep neural network [2]. This method has been proven to successfully synthesize complex scenes with high fidelity. Scene Representation Networks (SRNs) also use continuous volumetric scene representations to generate images from various viewpoints [12]. SRNs combine the advantages of geometry-based and image-based rendering by learning a continuous 3D representation and differentiable rendering for synthesizing novel views. Deep-Voxels utilizes a voxel-based representation, mapping from 3D world coordinates to 2D images, enabling view synthesis with the same appearance and geometry [13].

2.2 *Geometric Neural Rendering*

Geometric neural rendering focuses on generating 3D models and scenes by learning to map input data to 3D structures or depth maps. These techniques often involve the use of differentiable rendering algorithms to create realistic images from the generated 3D structures. One example of geometric neural rendering is Differentiable Ray Tracing (DRT), which introduces a differentiable version of the classic ray tracing algorithm, enabling gradient-based optimization of scene parameters [14]. DRT allows for the simultaneous optimization of geometry, materials, and lighting by using a differentiable approximation of the rendering equation. Neural Mesh Renderer (NMR) is another approach that uses neural networks to learn a differentiable rendering function, allowing for gradient-based optimization of mesh geometry and texture [15]. NMR combines a differentiable rasterizer with a shading model, enabling the end-to-end optimization of 3D object reconstruction tasks. The Soft

Rasterizer technique offers a differentiable rendering pipeline for generating images from 3D models, which can be used to optimize model parameters through gradient descent [16]). By approximating the rasterization process using a soft visibility function, Soft Rasterizer enables the direct optimization of 3D model parameters in a differentiable manner.

3 Generative Models

3.1 *Brief History of Generative Adversarial Networks*

Generative Adversarial Networks are a subcategory of machine learning frameworks designed to generate new, previously unseen data samples [3]. The initial paper introduced GANs and demonstrated their potential for generating synthetic images. The results showed that GANs could generate high-quality samples compared to other generative models at that time. Radford et al. [17] introduced Deep Convolutional GANs (DCGANs), which utilized convolutional neural networks (CNNs) in both the generator and discriminator. This architecture improved image generation quality and enabled stable training. Kingma et al. [4] proposed a Variational Autoencoders (VAEs) model that learn to represent data in a low-dimensional latent space. VAEs provide a principled approach to learning latent representations in an unsupervised manner. Their flexible architecture and smooth latent space make them suitable for a wide range of applications in machine learning and artificial intelligence.

Arjovsky et al. [18] proposed Wasserstein GAN (WGAN), which addressed the problem of mode collapse and unstable training by introducing a new loss function based on the Wasserstein distance. This modification improved the training stability and the quality of generated samples. Karras et al. [19] introduced a training methodology called Progressive Growing of GANs, which enabled the generation of high-resolution images by gradually increasing the resolution during training. This approach significantly improved the quality of generated images.

Brock et al. [20] proposed BigGAN, which scaled up GANs by using a larger architecture and more training data. BigGAN achieved state-of-the-art results in generating high-quality, high-resolution images. Karras et al. [21] introduced several innovations that further improved the quality and controllability of GAN-generated images. These include the incorporation of style-based generator architecture, adaptive instance normalization, and the elimination of batch normalization.

3.2 *Basic GAN Architecture*

The GAN architecture is composed of two neural networks: a generator and a discriminator (illustrated in Fig. 2). The generator takes a random noise vector as an

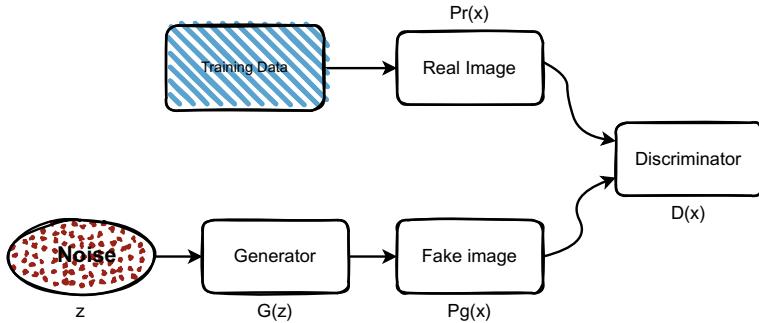


Fig. 2 Architecture of GAN [3]

input and outputs a sample which is intended to look like real data. The discriminator receives both the real data and the synthetic data from the generator, and tries to distinguish between them. These two networks are trained together in an adversarial manner, with the generator attempting to create samples that can fool the discriminator, and the discriminator working to recognize the real data from the fake.

The mathematics formulation of the GAN objective function can be expressed as follows:

$$\begin{aligned} \min_G \text{mix}_D V(D, G) = & E_{x \sim P_{\text{data}}(x)} [\log D(x)] \\ & + E_{z \sim P_z(z)} [\log(1 - D(G(z)))] \end{aligned} \quad (1)$$

where: $G(z)$ is the generator network that maps the random noise vector z to a synthetic sample. $D(x)$ is the discriminator network that maps a data sample x to a probability score indicating whether it is real or fake. $x \sim p(x)$ is the probability distribution of the real data. $z \sim p(z)$ is the probability distribution of the random noise vector z .

The objective function consists of two terms. The first term maximizes the probability that the discriminator correctly classifies the real data samples, while the second term maximizes the probability that the discriminator incorrectly classifies the synthetic data samples as real. The generator network is trained to minimize the second term by generating synthetic data that can fool the discriminator. During training, the two networks are updated alternatively (see Algorithm 1). First, the discriminator is trained to maximize $V(D, G)$ while keeping the generator fixed. Then, the generator is trained to minimize $V(D, G)$ while keeping the discriminator fixed. This alternating training process continues until both networks converge to an equilibrium point, when the discriminator is unable to tell the difference between the real and synthetic data because the generator creates synthetic data that is identical to the real data.

Algorithm 1 Training a Generative Adversarial Network (GAN)

```

1: Input: Number of training epochs  $n_{\text{epochs}}$ , batch size  $m$ , learning rate  $\eta$ 
2: Initialize generator  $G$  and discriminator  $D$  with random weights
3: Define the generator loss function:  $\mathcal{L}_G = -\mathbb{E}_{z \sim p_z(z)}[\log D(G(z))]$ 
4: Define the discriminator loss function:  $\mathcal{L}_D = -\mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] - \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$ 
5: for epoch = 1, 2, ...,  $n_{\text{epochs}}$  do
6:   for step = 1, 2, ...,  $\frac{n_{\text{samples}}}{m}$  do
7:     Sample a minibatch of  $m$  real data samples  $\{x^{(1)}, \dots, x^{(m)}\} \sim p_{\text{data}}(x)$ 
8:     Sample a minibatch of  $m$  noise samples  $\{z^{(1)}, \dots, z^{(m)}\} \sim p_z(z)$ 
9:     Update the discriminator weights by performing one step of gradient ascent on  $\mathcal{L}_D$ :

$$\theta_D \leftarrow \theta_D + \eta \nabla_{\theta} \mathcal{L}_D$$

10:    Update the generator weights by performing one step of gradient ascent on  $\mathcal{L}_G$ :

$$\theta_G \leftarrow \theta_G + \eta \nabla_{\theta} \mathcal{L}_G$$

11:   end for
12: end for
13: Output: Trained generator  $G$  and discriminator  $D$ 

```

3.3 GAN-Based Methods for Image Synthesis

The field of image rendering has undergone a revolution due to generative adversarial networks, providing a variety of methods for generating realistic images. This section presents a review of various GAN-based image synthesis techniques.

3.3.1 Variational Autoencoders (VAEs)

VAEs are a category of the deep generative models known as variational autoencoders, that have gained significant attention due to their ability to learn meaningful latent representations of data and generate new samples. Introduced by Kingma et al. [4], VAEs combine deep neural networks with probabilistic Bayesian inference to provide a powerful framework for unsupervised learning. An encoder (or inference network) maps input data to a latent space in the VAE architecture, and a decoder (or generative network) reconstructs the data from the latent space. The latent space is modeled using a probabilistic distribution, typically a Gaussian distribution, which enforces a structured continuous representation of the data.

A key innovation of VAEs is the use of a variational inference technique to optimize the model. This approach involves maximizing the Evidence Lower Bound (ELBO), which consists of two terms: the reconstruction loss and the Kullback-Leibler (KL) divergence. The reconstruction loss measures how well the decoder reconstructs the input data, while the KL divergence encourages the learned latent distribution to be close to a predefined prior distribution (e.g., a standard Gaussian

distribution). Balancing these two terms allows VAEs to learn compact and semantically meaningful latent representations. Let x denote the data, z denote the latent variables, and $p_\theta(x, z)$ represent the joint distribution modeled by the VAE, where θ are the generative model parameters. Let $q_\phi(z|x)$ be the approximate posterior, where ϕ are the recognition model parameters.

The objective function (ELBO) of the VAE is given by:

$$\mathcal{L}(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \text{KL}(q_\phi(z|x)||p_\theta(z)) \quad (2)$$

Here, $\log p_\theta(x|z)$ is the reconstruction loss, which measures how well the VAE can reconstruct the input data from the latent variables, and $\text{KL}(q_\phi(z|x)||p_\theta(z))$, which operates as a regularisation term, is the Kullback-Leibler divergence between the approximate posterior $q_\phi(z|x)$ and the prior $p_\theta(z)$.

The objective is to maximise the ELBO with respect to the parameters of the generative and recognition models, θ and ϕ :

$$\theta^*, \phi^* = \arg \max_{\theta, \phi} \mathcal{L}(\theta, \phi; x) \quad (3)$$

3.3.2 Progressive Growing of GANs

Progressive growing of GANs is a technique used to train generative adversarial networks (GANs) for improved image quality and stability. The technique was introduced in [19]. The main idea behind progressive growing is to start training the GAN on low-resolution images and gradually increase the resolution as training progresses. This allows the GAN to learn the low-level features first and then build upon them to learn more complex features. The process involves adding layers to both the generator and discriminator networks, and gradually increasing the resolution of the input and output images.

By starting with low-resolution images, the GAN is able to learn basic shapes and structures before moving on to more complex details. This results in images with more realistic and detailed features. Additionally, the progressive growing approach helps to stabilize the training process by avoiding mode collapse, which is a common problem in GANs where the generator only learns to produce a limited set of images. One of the main advantages of progressive growing is its ability to generate high-resolution images that are visually appealing and realistic (see Fig. 3). This has applications in a variety of fields, including computer graphics, art, and fashion.

3.3.3 Style-Based Generator Architecture

The StyleGAN architecture, which is also known as [8], is a type of generative adversarial network (GAN) created by Karras et al. Its purpose is to differentiate the style and content of images and to give more exact control over the image generation



Fig. 3 A collection of images generated using the CELEBA-HQ dataset, showcasing the high-resolution and photorealistic quality achieved by the model [19]

process. The architecture consists of a generator network and a discriminator network. The generator network takes in a random noise vector and outputs an image, while the discriminator network receives an image as input and provides a probability that it is real rather than fake.

StyleGAN is composed of a generator network made up of several convolutional layers that steadily increase the image's resolution. Unlike the usual GANs, the results of each convolutional layer are not fed straight into the next layer; instead, intermediate latent vectors are inserted in between. These latent vectors control the look of the image at different levels of abstraction, offering a more precise control over the creation process. To further improve this process, StyleGAN has a brand new mapping network that transforms random noise into the intermediate latent vectors used by the generator network. This mapping network is trained to make sure that these latent vectors possess a meaningful semantic representation which can be altered easily in order to manipulate the style of the generated images.

StyleGAN has been shown to produce state-of-the-art results in a variety of image synthesis tasks, including generating high-resolution images of faces and objects. Its ability to separate the style and content of images has also made it a popular choice for image editing applications.

In the case of StyleGAN, the generator is modified to incorporate style information. This is done by introducing a mapping network (F) and an adaptive instance normalization (AdaIN) operation. The mapping network takes the random noise vector z as input and generates an intermediate latent space w . This intermediate latent space w is then used to modulate the style of the generated image through the AdaIN operation. Figure 4 presents some results generated by the StyleGan model.

The generator objective function remains the same, but with the modified generator G' :



Fig. 4 This figure showcases a collection of images generated by the style-based generator trained on the FFHQ dataset. The style-based generator’s ability to generate such diverse and photorealistic images highlights the effectiveness of the model and its capacity for capturing the complex structure of human faces in high-resolution images [8]

$$L_{G'} = -E \left[\log(D(G'(F(z))) \right] \quad (4)$$

where $G'(F(z))$ represents the output of the generator given the intermediate latent space w .

3.3.4 Cycle-Consistent GANs for Unpaired Image-to-Image Translation

Cycle-Consistent GANs (CycleGANs) is a type of GANs that are used for unpaired image-to-image translation. It was first proposed by Jun et al. [22] as a way to learn mappings between two different image domains without requiring paired training data. CycleGAN’s fundamental concept is to use cycle consistency to assure that the generated images match the input image. Thus, the generator can learn to translate between image domains (see Fig. 5). The basic architecture of CycleGAN consists of two GANs: a generator G that maps images from one domain to another and a discriminator D that distinguishes between real and fake images. The objective function of CycleGAN consists of two parts: adversarial loss and cycle-consistency loss. The adversarial loss ensures that the generated images are indistinguishable from real images, and the cycle-consistency loss ensures that the translations are consistent when applying both generators sequentially.

Adversarial Loss (using the least-squares GAN formulation):

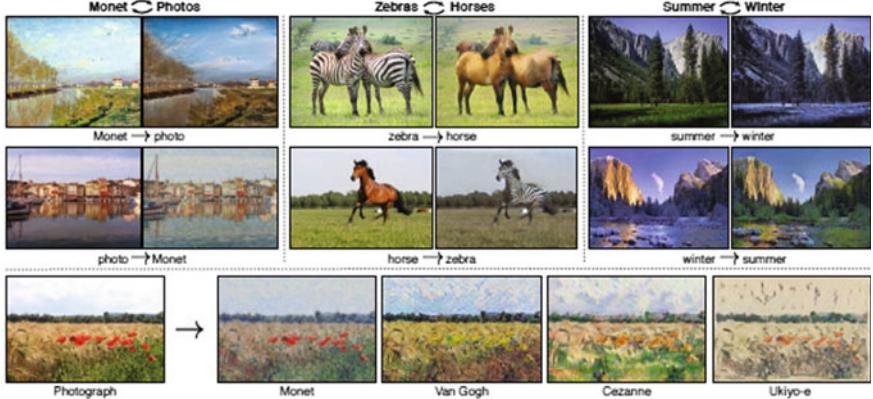


Fig. 5 CycleGAN achieve high-quality image-to-image translation [22]

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[(D_Y(y) - 1)^2] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[(D_Y(G(x)))^2] \quad (5)$$

$$\mathcal{L}_{\text{GAN}}(F, D_X, Y, X) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[(D_X(x) - 1)^2] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[(D_X(F(y)))^2] \quad (6)$$

Cycle-Consistency Loss:

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1] \quad (7)$$

Total Loss:

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F) \quad (8)$$

Here, λ is a hyperparameter that controls the importance of the cycle-consistency loss.

In order to train the CycleGAN, the generators and discriminators are alternately optimised using gradient descent to reduce the total loss:

$$G^*, F^* = \arg \min_{G, F} \max_{D_X, D_Y} \mathcal{L}(G, F, D_X, D_Y) \quad (9)$$

3.3.5 Stacked GANs for Generating High-Resolution Images

Zhang et al. [23] proposes a novel method for generating high-resolution images using stacked generative adversarial networks (GANs) in a two-stage process as shown in Fig. 6. Two stages comprise the proposed method: the first stage generates

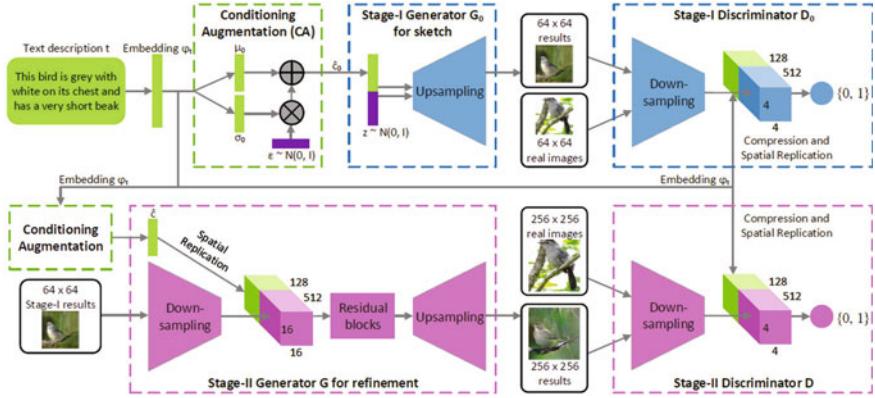


Fig. 6 Stage-I generator: the text embedding is combined with a random noise vector, and a low-resolution image is generated. Stage-I discriminator: the generated low-resolution image and the real image are concatenated with the text embedding. Stage-II generator: the low-resolution image from Stage-I and the text embedding are combined. Stage-II discriminator: the generated high-resolution image and the real image are concatenated with the text embedding [23]

a low-resolution image, and the second stage employs a high-resolution generator to convert the low-resolution image into a high-resolution image. In the first stage, a GAN is trained to generate a low-resolution image from a random noise vector.

The generator takes in a noise vector and produces a low-resolution image, while the discriminator takes in both the generated low-resolution image and a real low-resolution image and determines whether each is real or fake. The generator and discriminator are trained in an adversarial manner to generate realistic-looking low-resolution images. In the second stage, a high-resolution generator is trained to convert the low-resolution image generated in the first stage into a high-resolution image. The high-resolution generator takes in the low-resolution image and generates a high-resolution image, while the discriminator takes in both the generated high-resolution image and a real high-resolution image and determines whether each is real or fake. The high-resolution generator and discriminator are also trained in an adversarial manner to generate realistic-looking high-resolution images.

The mathematical formulas involved in the proposed method are similar to those used in traditional GANs, including the generator and discriminator loss functions, as well as the backpropagation algorithm used for training. The main difference is the use of two separate stages, each with their own generator and discriminator, to generate high-resolution images.

Stage-I GAN Loss Functions:

$$\begin{aligned} \mathcal{L}_{\text{GAN}}^{\text{I}}(G_1, D_1, T, X_1) = & \mathbb{E}_{x_1 \sim p_{\text{data}}(x_1), t \sim p_{\text{data}}(t)} [\log D_1(x_1, t)] \\ & + \mathbb{E}_{t \sim p_{\text{data}}(t), z \sim p_z(z)} [\log(1 - D_1(G_1(z, t), t))] \end{aligned} \quad (10)$$

$$\mathcal{L}_{\text{KL}}^{\text{I}}(G_1) = \mathbb{E}_{t \sim p_{\text{data}}(t), z \sim p_z(z)} [\text{KL}(N(\mu(z, t), \sigma(z, t)) || N(0, I))] \quad (11)$$

Stage-II GAN Loss Functions:

$$\begin{aligned}\mathcal{L}_{\text{GAN}}^{\text{II}}(G_2, D_2, T, X_2) = & \mathbb{E}_{x_2 \sim p_{\text{data}}(x_2), t \sim p_{\text{data}}(t)} [\log D_2(x_2, t)] \\ & + \mathbb{E}_{t \sim p_{\text{data}}(t), x_1 \sim G_1(z, t)} [\log(1 - D_2(G_2(x_1, t), t))]\end{aligned}\quad (12)$$

$$\mathcal{L}_{\text{KL}}^{\text{II}}(G_2) = \mathbb{E}_{t \sim p_{\text{data}}(t), x_1 \sim G_1(z, t)} [\text{KL}(N(\mu(x_1, t), \sigma(x_1, t)) || N(0, I))] \quad (13)$$

Total Loss:

$$\begin{aligned}G_1^*, G_2^*, D_1^*, D_2^* = \arg \min_{G_1, G_2} \max_{D_1, D_2} (\mathcal{L}_{\text{GAN}}^{\text{I}}(G_1, D_1, T, X_1) + \mathcal{L}_{\text{GAN}}^{\text{II}}(G_2, D_2, T, X_2) \\ + \lambda_1 \mathcal{L}_{\text{KL}}^{\text{I}}(G_1) + \lambda_2 \mathcal{L}_{\text{KL}}^{\text{II}}(G_2))\end{aligned}\quad (14)$$

Here, λ_1 and λ_2 are hyperparameters controlling the importance of the KL divergence regularization terms.

3.3.6 BigGAN for High-Fidelity Image Synthesis

BigGAN [20] is a paper that introduces an improved version of the GAN architecture which allows for efficient large-scale training and accurate image synthesis. The BigGAN model is based on the conditional GAN architecture, with a generator and a discriminator network. The generator, given a random noise vector and a class descriptor, produces an image belonging to that specific class. The discriminator, on the other hand, will take an image and its corresponding class label as input and outputs whether it is real or fake.

The main contribution of BigGAN is the use of a hierarchical generator architecture, which consists of multiple generator modules that operate at different spatial resolutions. The generator modules are trained separately, and their outputs are combined to produce the final image. This approach allows for the generation of high-resolution images (up to 512×512 pixels) with fine-grained details.

BigGAN incorporates a truncation trick when training that truncates the generator's latent space to focus on the essential dimensions, which boosts the variety of produced images and forestalls mode collapse. Mathematically, the loss functions for the generator and discriminator networks in BigGAN are analogous to those of the original GAN. The generator network is optimized to minimize binary cross-entropy loss, while the discriminator network is optimized to maximize the same loss. Additionally, BigGAN utilizes a feature matching loss to ensure that generated images match the characteristics of real images at multiple layers of the discriminator network.

BigGAN represents a significant advancement in GAN research, enabling the generation of high-quality, high-resolution images at a large scale as shown in Fig. 7.



Fig. 7 Some results generated by BigGAN model [20]

BigGAN uses the hinge loss for its adversarial loss functions. Let's define the generator G , discriminator D , real data samples x , class labels y , random noise z , and generated samples $G(z, y)$.

Generator Loss:

$$\mathcal{L}_G(G, D) = -\mathbb{E}_{z \sim p_z(z), y \sim p_{\text{data}}(y)}[D(G(z, y), y)] \quad (15)$$

Discriminator Loss:

$$\begin{aligned} \mathcal{L}_D(G, D) = & \mathbb{E}_{x \sim p_{\text{data}}(x), y \sim p_{\text{data}}(y)}[\max(0, 1 - D(x, y))] \\ & + \mathbb{E}_{z \sim p_z(z), y \sim p_{\text{data}}(y)}[\max(0, 1 + D(G(z, y), y))] \end{aligned} \quad (16)$$

Total Loss:

$$G^*, D^* = \arg \min_G \max_D (\mathcal{L}_G(G, D) + \mathcal{L}_D(G, D)) \quad (17)$$

4 Applications of Generative Models for Image Rendering

Neural rendering refers to the process of synthesizing images or videos using deep learning techniques. GAN-based models have been used to achieve impressive results in various neural rendering applications. Particularly global illumination, this last is a key aspect of realistic image synthesis, as it accounts for the complex interplay of light in a scene, including direct lighting, indirect lighting, and reflections. Generative

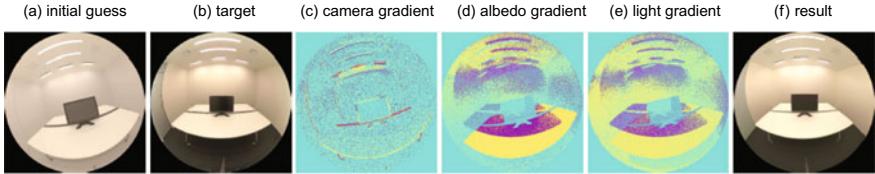


Fig. 8 The model successfully generates an image that closely resembles the photo reference [24]

models, particularly deep learning approaches, have been used to tackle various aspects of global illumination in image rendering.

Tzu et al. [24] present a novel approach to differentiate Monte Carlo ray tracing, a popular rendering algorithm, through edge sampling (see Fig. 8). This technique enables the use of gradient-based optimization techniques for inverse rendering problems, such as material estimation, lighting design, and camera parameter optimization. The authors propose an efficient method to compute gradients of Monte Carlo estimators with respect to scene parameters, overcoming challenges associated with the discontinuities and high-dimensional integrals that are typically present in ray tracing.

The main contribution of this work is a new gradient sampling technique called Edge Sampling. This method samples discontinuity edges in the integrand of the rendering equation and computes the gradients of the Monte Carlo estimator with respect to the input parameters by taking into account the contribution of the discontinuity edges. This paper introduces a differentiable Monte Carlo ray tracing method through Edge Sampling, which opens up new possibilities for gradient-based optimization in inverse rendering problems and improves the performance and capabilities of existing rendering algorithms.

Deep Shading [25] is a novel deep learning-based method for generating realistic shadows in scenes illuminated by a single light source. This paper presents a framework that learns illumination for generative shadow rendering, addressing the challenges of simulating accurate and natural-looking shadows. The proposed approach relies on a conditional generative adversarial network (cGAN) to model the relationship between illumination and shadows.

The framework consists of two main components: an illumination estimation network and a shadow rendering network (see Fig. 9). The illumination estimation network predicts the position and intensity of the light source in a given input image. The shadow rendering network then generates realistic shadows based on the estimated light source and a given 3D model.

The paper [13] introduce a two-step training process for the illumination estimation network. In the first step, the network is trained using synthetic images with known ground truth illumination. In the second step, the network is fine-tuned using real-world images with estimated illumination obtained from an optimization-based method. This approach allows the model to generalize well to real-world scenes. The authors propose a continuous volumetric scene function that maps 3D world coordi-

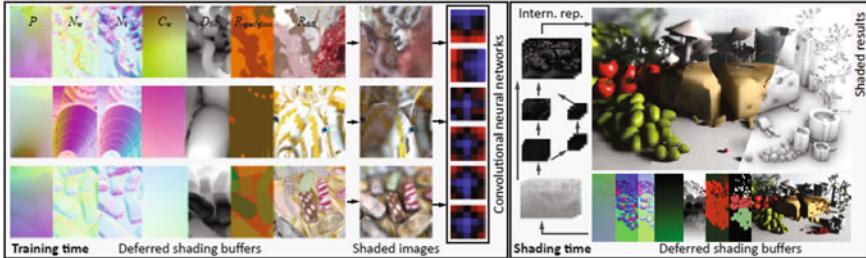
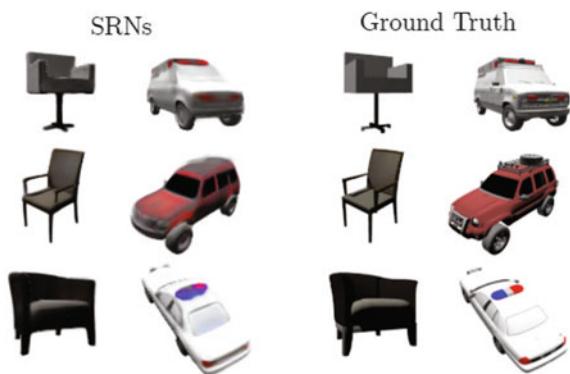


Fig. 9 Illustration of how the proposed method utilizes a CNN to learn a mapping from deferred shading buffer attributes to RGB colors during the training phase, and then applies this learned mapping at run-time to generate realistic images with various effects at interactive rates [25]

Fig. 10 Qualitative comparisons between ground truth and some results generated by [13]



nates (x , y , z) to a feature vector, representing a scene. To learn this scene function, they use a combination of an encoder network and a decoder network. The encoder is responsible for learning a latent code for a given scene, while the decoder maps the 3D coordinates and latent code to a feature representation. The final output is rendered using a differentiable volume rendering layer, which enables the network to be trained end-to-end.

The Scene Representation Networks (SRNs) approach is designed to handle a variety of 3D structures, including complex geometries and multi-view consistency as shown in Fig. 10. The authors show that SRNs are more effective than previous methods for several tasks, such as novel view synthesis, view interpolation, and 3D reconstruction. They also demonstrate the robustness of SRNs to different input modalities, like 2D images and 3D point clouds.

DeepLight [26] presents a deep learning-based approach to estimate and render plausible shadows in 2D images, given a single input image without any depth information (see Fig. 11). The authors propose a two-stage pipeline that consists of an illumination estimation network and a shadow rendering network.

The first stage, the illumination estimation network, predicts the illumination conditions of the scene. It is trained using a novel, unsupervised approach that relies

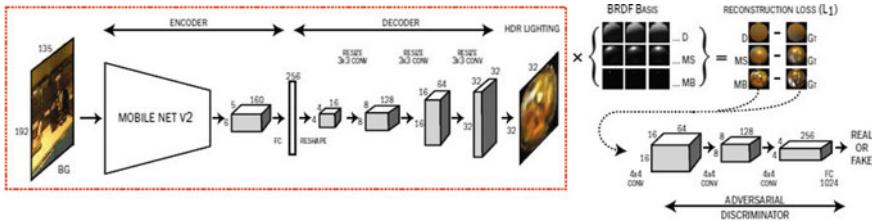


Fig. 11 DeepLight architecture [26]

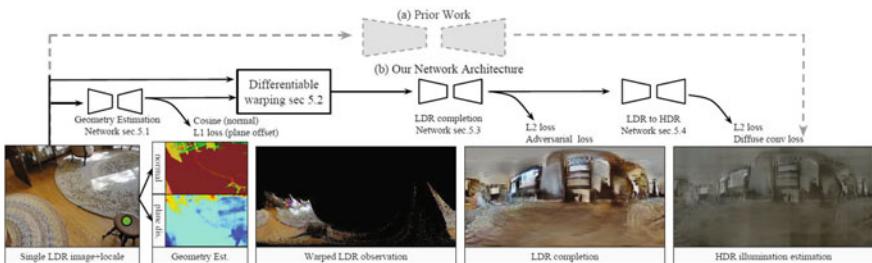


Fig. 12 Overview of the model neural illumination [27]

on a shadow consistency loss and a shadow-aware structure similarity index. This network is responsible for estimating the illumination direction and the light source intensity in the scene, which are then used as input for the shadow rendering network.

The second stage, the shadow rendering network, generates shadow masks based on the input image and the estimated illumination conditions. The network is trained using a supervised approach with paired data consisting of images and their corresponding shadow masks. A multi-scale loss function is used to train this network, which helps generate high-quality shadows by preserving both global and local shadow structures. Neural Illumination (NI) [27] presents a novel deep learning approach to estimate the lighting conditions of indoor scenes. The authors propose a method which is designed to predict the spatially-varying illumination from a single, low-dynamic-range (LDR) image of an indoor environment.

The key contribution of this work is the introduction of a new architecture, as shown in Fig. 12 that combines a neural radiance field (NeRF) with a spherical harmonics (SH) basis representation of the light field. This combination enables the model to capture both the geometric structure and the lighting information of the scene. The authors train the NI model using a large dataset of indoor scenes with illumination data, which they obtain through a novel data collection pipeline.

The paper demonstrates that the proposed NI model outperforms state-of-the-art methods in predicting lighting conditions from a single LDR image. Furthermore, the authors show that the NI model can be used in various practical applications, such as relighting objects in the scene, estimating the lighting for augmented reality, and enhancing images by adjusting their illumination.

5 Challenges and Future Directions

Global illumination is a key aspect of realistic image synthesis, as it accounts for the complex interplay of light in a scene, including direct lighting, indirect lighting, and reflections. Generative models, particularly deep learning approaches, have been used to tackle various aspects of global illumination in neural rendering. Some of these applications include:

Direct and Indirect Lighting Estimation: Generative models can be employed to predict direct and indirect lighting components in a scene. For example, the Deep-Light framework uses a deep learning model to learn a global illumination estimator that can predict both direct and indirect lighting from an input scene description.

Reflectance and Material Modeling: Generative models can be used to synthesize material properties and reflectance functions for virtual objects, allowing for more accurate and realistic rendering of light interactions with various surfaces. Examples include the use of GANs and Variational Autoencoders (VAEs) for generating material properties and bidirectional reflectance distribution functions (BRDFs).

Light Transport Simulation: Generative models can be used to learn efficient approximations of light transport, replacing traditional ray-tracing or photon mapping techniques with more efficient and fast neural network-based solutions. One example is the Neural Path Tracer, which uses a neural network to learn an approximation of the light transport equation in a given scene.

Image-based Relighting: Generative models can be employed for image-based relighting, enabling users to change the lighting conditions of an existing image without the need for a full 3D scene reconstruction. Deep Relightable Appearance Models and Neural Illumination are examples of deep learning approaches for image-based relighting.

Environment Map Estimation: Generative models can be used to estimate environment maps from a single image or a small set of images, allowing for the rendering of objects in different lighting conditions. The Deep Environment Map Estimation (DEME) framework uses a deep learning model to learn environment maps from input images.

Real-time Global Illumination: Generative models can be employed to approximate global illumination in real-time, enabling interactive applications such as video games and virtual reality to achieve more realistic lighting effects without sacrificing performance. One example is the Spherical Gaussians (SG) representation, which can be used to approximate complex lighting environments with a small number of parameters, enabling real-time global illumination calculations.

These applications of generative models for global illumination neural rendering showcase the potential of deep learning techniques to improve the realism, efficiency, and interactivity of image synthesis, with impacts on computer graphics, animation, and immersive technologies.

6 Conclusion

In conclusion, GAN-based image rendering has revolutionized the field of computer graphics and computer vision by enabling the generation of realistic and high-quality images. The ability to produce images that cannot be distinguished from reality has opened up a world of possibilities for applications such as virtual and augmented reality, image restoration and manipulation, and even the creation of new art forms.

Despite the success of GANs, there are still challenges that need to be addressed, such as improving the stability and convergence of training, addressing mode collapse, and improving the ability to generate diverse and controllable outputs. Additionally, there is still much to explore in terms of new applications of GANs, such as in medical imaging and scientific data analysis. As GAN research continues to evolve, we can expect to see even more exciting advancements in the field of image rendering. With the development of new architectures, loss functions, and training techniques, GANs will continue to push the boundaries of what is possible in terms of image synthesis and manipulation.

References

- Thies, J., Zollhöfer, M., & Nießner, M. (2019). Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4), 1–12.
- Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2021). NERF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1), 99–106.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139–144.
- Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint [arXiv:1312.6114](https://arxiv.org/abs/1312.6114)
- Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., & Graves, A. (2016). Conditional image generation with PixelCNN decoders. *Advances in Neural Information Processing Systems*, 29.
- LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., & Jackel, L. (1989). Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, 2.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4401–4410).
- Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2414–2423).
- Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., & Metaxas, D. N. (2018). Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8), 1947–1962.
- Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125–1134).

12. Sitzmann, V., Thies, J., Heide, F., Niessner, M., Wetzstein, G., & Zollhöfer, M. (2019). DeepVoxels: Learning persistent 3D feature embeddings. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2437–2446).
13. Sitzmann, V., Zollhöfer, M., & Wetzstein, G. (2019). Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32.
14. Loubet, G., Holzschuch, N., & Jakob, W. (2019). Reparameterizing discontinuous integrands for differentiable rendering. *ACM Transactions on Graphics (TOG)*, 38(6), 1–14.
15. Kato, H., Ushiku, Y., & Harada, T. (2018). Neural 3d mesh renderer. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3907–3916).
16. Liu, S., Li, T., Chen, W., & Li, H. (2019). Soft rasterizer: A differentiable renderer for image-based 3D reasoning. In *Proceedings of the IEEE international conference on computer vision* (pp. 7708–7717).
17. Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint [arXiv:1511.06434](https://arxiv.org/abs/1511.06434)
18. Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein GAN. arXiv preprint [arXiv:1701.07875](https://arxiv.org/abs/1701.07875)
19. Karras, T., Aila, T., Laine, S., Lehtinen, J. (2017). Progressive growing of GANs for improved quality, stability, and variation. arXiv preprint [arXiv:1710.10196](https://arxiv.org/abs/1710.10196)
20. Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale GAN training for high fidelity natural image synthesis. arXiv preprint [arXiv:1809.11096](https://arxiv.org/abs/1809.11096)
21. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of styleGAN. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8110–8119).
22. Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223–2232).
23. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., & Metaxas, D. N. (2017). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 5907–5915).
24. Li, T. M., Aittala, M., Durand, F., & Lehtinen, J. (2018). Differentiable Monte Carlo ray tracing through edge sampling. *ACM Transactions on Graphics (TOG)*, 37(6), 1–11.
25. Nalbach, O., Arabadzhyska, E., Mehta, D., Seidel, H. P., & Ritschel, T. (2017, July). Deep shading: Convolutional neural networks for screen space shading. *Computer Graphics Forum*, 36(4), 65–78.
26. LeGendre, C., Ma, W. C., Fyffe, G., Flynn, J., Charbonnel, L., Busch, J., & Debevec, P. (2019). DeepLight: Learning illumination for unconstrained mobile mixed reality. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5918–5928).
27. Song, S., & Funkhouser, T. (2019). Neural illumination: Lighting prediction for indoor environments. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6918–6926).

Dsmk-DcSeg-Lap, a Generative Adversarial Network Guided by Dark-Chanel and Segmentation to Smoke Removal in Laparoscopic Images



Hugo Moreno, Sebastián Salazar-Colores, Luis M. Valentín,
and Gerardo Flores

Abstract In this chapter, a computational approach is proposed to address the challenge of degraded visibility caused by smoke during laparoscopic surgery. The visualization of organs and tissues is hindered by the presence of smoke, which results from dissection tools. This, in turn, leads to potential errors and increased surgical duration, ultimately impacting patient outcomes. To overcome this issue, a novel neural architecture is introduced, which consists of two autoencoders trained using the generative neural network paradigm. The image segmentation on the laparoscopic image is performed by the first autoencoder, while the second autoencoder incorporates this segmented image as an additional fifth channel. To evaluate the effectiveness of the approach, comprehensive quantitative assessments are conducted, and the results are compared with state-of-the-art desmoking and dehazing techniques. Performance evaluation is carried out using commonly used metrics in the field. The superiority of the proposed method over existing approaches is demonstrated by the obtained results. This makes the method highly suitable for integration into medical systems using embedded devices.

Keywords Conditional generative adversarial network · Dark channel · Semantic segmentation · Laparoscopy · Desmoking

H. Moreno · S. Salazar-Colores · G. Flores

Laboratorio de Percepción y Robótica [LAPyR], Center for Research in Optics, Loma del Bosque 115, León, Guanajuato 37150, Mexico

e-mail: hugoamj@cio.mx

S. Salazar-Colores

e-mail: sebastian.salazar@cio.mx

L. M. Valentín

Centro de Investigaciones en Óptica, A.C. Unidad Aguascalientes, Aguascalientes 20200, Mexico
e-mail: luismvc@cio.mx

G. Flores (✉)

Department of Electrical and Computer Engineering, University of Texas at Dallas, Richardson TX 75080, USA

e-mail: gflores@cio.mx; gxf230006@utdallas.edu

1 Introduction

Laparoscopy is a minimally invasive surgical procedure that has become increasingly popular in recent years. It offers a number of advantages over traditional open surgery, including shorter recovery times, less pain, and smaller scars. However, one of the challenges of laparoscopy is the presence of smoke, which can obscure the surgeon's view of the surgical field. Smoke is generated during laparoscopy by a number of sources, including the use of electrocautery to cut and seal tissue. Smoke can impair the surgeon's visibility and make it difficult to identify and operate on small structures. It can also increase the risk of infection by carrying bacteria and other pathogens into the surgical field.

A variety of methods have been developed to remove smoke during laparoscopy. These include manual smoke evacuation, smoke filters, gas flow evacuation systems, and advanced vision technologies. Manual smoke evacuation is the most basic method and involves the use of a suction probe to remove smoke from the surgical field. Smoke filters are devices that are attached to the laparoscope and trap smoke particles, allowing the gas to escape. Gas flow evacuation systems use a continuous stream of gas to flush smoke out of the surgical field. Advanced vision technologies use image processing algorithms to remove smoke from the surgeon's view.

While these methods can be effective in removing smoke, they can also be time-consuming and cumbersome. Manual smoke evacuation can be tiring for the surgeon and assistant, and smoke filters can clog and need to be replaced frequently. Gas flow evacuation systems can create a strong wind that can make it difficult to control the laparoscope, and advanced vision technologies can be expensive and complex. As a result of these limitations, there is a need for a more efficient and effective method of smoke removal for laparoscopy. A recent systematic review and meta-analysis [1] found that smoke evacuation systems were effective in reducing smoke levels during laparoscopy. However, the authors noted that more research is needed to determine the optimal type of smoke evacuation system for different types of laparoscopic procedures. Another recent study [2] found that smoke management was a significant factor in the risk of infection during laparoscopy. The authors found that surgeons who used smoke evacuation systems had a lower risk of infection than those who did not use smoke evacuation systems. A third recent study [3] found that novel smoke evacuation systems were effective in reducing smoke levels and improving visibility during laparoscopy. The authors concluded that novel smoke evacuation systems could be a valuable tool for improving the safety and efficiency of laparoscopy.

2 Related Work

Smoke removal from images has been a challenge for researchers for a long time. Before the emergence of artificial intelligence (AI), several traditional methods were developed and used for this purpose. These approaches included filtering techniques, image dehazing, and physical modeling [4–6]. Although they achieved some degree

of success, these techniques often had limitations in terms of image quality and adaptability to different smoke conditions.

However, with the advent of AI, more advanced and effective approaches have emerged for smoke removal in images. These methods harness the power of machine learning algorithms and neural networks to achieve impressive results. The following are some state-of-the-art techniques used today:

- **Traditional Methods for Smoke Removal:** Before the emergence of AI, various traditional methods were developed and utilized for the removal of smoke from images. Included among these methods were filtering techniques, image dehazing, and physical modeling. Despite achieving some degree of success, these approaches often exhibited limitations in terms of image quality and adaptability to different smoke conditions [4–6].
- **AI Approaches for Smoke Removal:** With the advent of AI, more advanced and effective approaches have emerged for smoke removal in images. State-of-the-art methods and techniques include:
 - **Deep Learning Techniques:** Deep learning-based approaches, such as deep neural networks and autoencoders, have proven to be effective for smoke removal. These techniques learn intrinsic image characteristics and utilize them to separate smoke from the actual content [7, 8].
 - **Convolutional Neural Networks (CNNs):** CNNs have been utilized for processing smoky images. Specific smoke features and patterns are learned by these networks to remove smoke and enhance image quality [8].
 - **Generative Adversarial Networks (GANs):** GANs have been employed to generate clear images from smoky ones. These networks consist of a generator network and a discriminator network working together to produce realistic smoke-free images [9].
 - **Physics-Based Methods:** Certain approaches combine physics-based models with AI techniques to eliminate smoke from images. By leveraging knowledge about light scattering and absorption in the presence of smoke, these methods enhance image quality [10, 11].

The remaining sections of this chapter are structured as follows. Section 3 provides the background information, introducing the fundamental concepts and theories that form the basis of our proposed method. Section 4 introduced the dark channel, semantic segmentation and smoke simulation methodologies. Section 5 presents our proposed method, detailing its design, implementation, and any relevant techniques or algorithms utilized. Additionally, this section includes a description of the experimental setup. Section 6 describes the metrics used for the evaluation of the proposed method to compare with five state-of-the-art methods. Section 7 presents the results obtained. Furthermore, a qualitative comparative analysis of the methods. Finally, Section 8 concludes the chapter by summarizing the findings, highlighting the contributions of our proposed method, and discussing potential avenues for future research.

3 Background

In this section, we provide an overview of the key terms and concepts necessary to understand the research conducted in this study. The following terms will be discussed: Artificial Neural Network (ANN), Convolutional Layer, ReLU (Rectified Linear Unit), Pooling Layers, U-Net, Conditional GAN, and Pix2Pix.

These terms play a crucial role in the research methodology and techniques employed. Understanding their definitions and functionalities is essential for comprehending the subsequent sections of the paper.

3.1 Artificial Neural Network

An Artificial Neural Network (ANN) is a series of operations that takes a large amount of data as input and predicts an output with a certain probability of accuracy. Convolutional Neural Networks (CNNs), a type of ANN, are particularly effective at efficiently detecting and learning patterns in multidimensional data, making them highly valuable in image processing and computer vision tasks [12]. In the context of image inputs, CNN layers consist of neurons arranged in three dimensions: width, height, and depth. Each neuron is connected to a small region of the preceding layer [13].

To construct a CNN, three main types of layers are employed: Convolutional layers, Pooling layers, and Fully-Connected layers. These layers progressively transform the input image, from the original pixel values to the final class scores.

3.1.1 Convolutional Layer

A convolutional layer is responsible for performing convolution operations on the input data to extract relevant features. Convolution is achieved by sliding a set of filters over the input image, producing feature maps that highlight different aspects of the image [12].

The convolution operation in a convolutional layer can be represented by the following equation:

$$y[i, j] = \sigma \left(\sum_m \sum_n x[i + m, j + n] \cdot w[m, n] + b \right) \quad (1)$$

In this equation: $y[i, j]$ represents the output value at position (i, j) of the convolutional layer. $x[i + m, j + n]$ denotes the input value at position $(i + m, j + n)$. $w[m, n]$ denotes the weights (filters) corresponding to position (m, n) . b is the bias. σ is the activation function, which introduces nonlinearity to the output.

The convolution operation involves multiplying the input values $x[i + m, j + n]$ by the weights $w[m, n]$, summing these products, and then adding the bias b . Finally, the output is obtained by applying the activation function σ to introduce nonlinearity.

3.1.2 ReLU

The Rectified Linear Unit (ReLU) is an activation function frequently used in CNNs. It is defined as [14]:

$$Y_i = \max(0, Y_i) \quad (2)$$

ReLU offers several advantages over other activation functions, such as the hyperbolic tangent or traditional logistic functions. It efficiently propagates gradients, addressing the issue of vanishing gradients. ReLU sets negative values to zero, alleviating the problem of cancellation. Additionally, ReLU involves simple computations, making it an efficient implementation in CNNs.

3.1.3 Pooling Layers

Pooling layers are employed to reduce the spatial size of activation maps. Typically used after multiple layers, they reduce the computational requirements as data progresses through the network. These layers preserve important features while discarding non-significant data by reducing the spatial resolution [15].

Pooling layers operate by defining a window of a specific size and reducing the data within that window to a single value. The window then moves a certain number of positions, repeating the process until the entire activation volume is reduced. Two common methods of reduction are max pooling and average pooling. Max pooling selects the highest value within the window and discards the remaining values, while average pooling calculates the mean of the values within the window [15].

3.1.4 Softmax

After pooling, the images become smaller. To obtain a final classification, a softmax function is typically implemented when dealing with more than two classes [12]. To accommodate the softmax function, a fully connected layer is inserted between the last convolutional/pooling layer and the final prediction [13]. This involves rearranging each image in the stack into a simple vector, which is then fully connected to a layer that serves as the input for the softmax function.

The softmax function calculates the probability of an image belonging to each class. It takes the vector of rearranged values as input and converts each element into a probability. The final result, after iterating over all neurons, is a set of probabilities corresponding to each class [12].

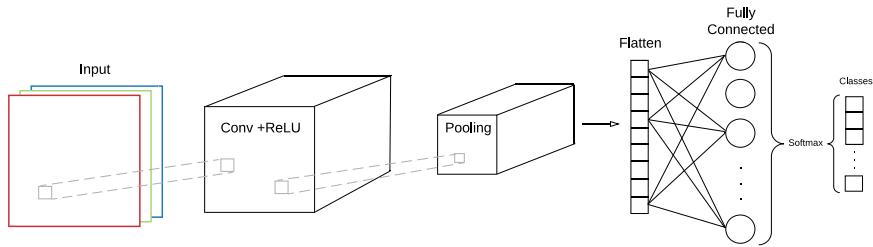


Fig. 1 A graphical and general representation of the sequence of layers and spacial dimensions of a traditional CNN where it can be seen that the input is a three dimensional (RGB channels) squared image and the output after the softmax function is the classification in terms of probability

$$\sigma_i(\alpha) = \frac{e^{\alpha_i}}{\sum_j e^{\alpha_j}} \quad (3)$$

Equation 3 represents the mapping of each vector, which represents an image from the previous stack, to a probability distribution. The left side of the equation denotes the probability of class i , taking the entire list of rearranged values as input, while j represents the position of that vector.

In Fig. 1, a general graphical representation of the architecture of a traditional CNN is provided. It illustrates the sequence of layers and spatial dimensions, demonstrating that the input is a three-dimensional (RGB channels) square image, and the output after the softmax function is the classification in terms of probability.

3.2 U-Net

The U-Net is a convolutional neural network architecture that was specifically designed for semantic segmentation tasks in computer vision. It was introduced by Ronneberger et al. in 2015 [16], initially for biomedical image segmentation but has since been widely adopted in various fields.

The U-Net architecture is characterized by a U-shaped encoder-decoder structure. The encoder path captures the context and extracts features from the input image through a series of convolutional and pooling layers, gradually reducing the spatial resolution while increasing the number of channels. The decoder path then uses upsampling and transposed convolutions to restore the spatial resolution, allowing for precise localization of features.

A distinctive feature of the U-Net is its skip connections. These connections allow information from the encoder path to be directly transmitted to the corresponding decoder path, facilitating the fusion of low-level and high-level features. This enables the network to capture both detailed local information and global context, enhancing the segmentation accuracy.

The U-Net has become widely popular due to its effectiveness in addressing the challenges of limited data and class imbalance in segmentation tasks. It has been successfully applied in various domains, including medical imaging, satellite imagery, and natural scene understanding.

Overall, the U-Net architecture provides a powerful tool for accurate and efficient semantic segmentation, making it an important reference in the field of computer vision.

3.3 Conditional GAN y Pix2Pix

Conditional Generative Adversarial Networks (cGANs) extend the standard Generative Adversarial Networks (GANs) framework by incorporating conditional information in the training process [17]. This enables the generation of realistic data samples based on specific input conditions or constraints.

The Pix2Pix model, introduced by Isola et al. [18] in 2016, is a notable example of a conditional GAN architecture. Pix2Pix is designed for image-to-image translation tasks, aiming to learn a mapping between an input image from one domain and a corresponding output image from another domain.

The Pix2Pix model consists of a generator network (G) and a discriminator network (D). The generator takes an input image (x) from one domain and generates a corresponding output image (y) in the target domain. The discriminator is trained to distinguish between the generated output images ($G(x)$) and the real output images (y) from the target domain. The generator and discriminator are trained adversarially, competing against each other to improve their respective performance.

The generator in Pix2Pix utilizes a U-Net architecture, which consists of an encoder path and a decoder path. The encoder captures the input image features, and the decoder generates the corresponding output image by upsampling and concatenating feature maps from the encoder. The U-Net structure allows for both local and global information integration, aiding in the generation of high-quality output images.

The objective of the Pix2Pix model is to minimize the adversarial loss (L_{adv}) and a reconstruction loss (L_{recon}). The adversarial loss encourages the generator to produce output images that are indistinguishable from real images, while the reconstruction loss ensures that the generated output image is similar to the ground truth image.

The overall objective function for training the Pix2Pix model can be represented as follows:

$$\min_G \max_D \mathcal{L}_{adv}(G, D) + \lambda \mathcal{L}_{recon}(G) \quad (4)$$

where $\mathcal{L}_{adv}(G, D)$ represents the adversarial loss, which is typically defined using the binary cross-entropy loss. $\mathcal{L}_{recon}(G)$ denotes the reconstruction loss, which measures

the difference between the generated output image and the ground truth image. λ is a hyperparameter that balances the importance of the adversarial and reconstruction losses.

By training the Pix2Pix model with paired examples of input and output images, it learns to generate output images that are visually similar to the ground truth images, while respecting the desired translation or conditioning specified by the input.

4 Dark-Channel and Segmentation

In this section, we will explore the dark-channel and segmentation methodologies employed and the advantages they offer to the designed network.

4.1 Hazy Image Formation

Images become degraded by smoke or vapors in the environment because light passing through these media is scattered and absorbed. This scattering and absorption of light leads to a reduction in visual quality, often resulting in partial or complete occlusion of elements within the image.

The formation of a hazy image, as shown in Fig. 2, can be mathematically modeled by taking into account the effects of light absorption and scattering as it travels through a cloudy environment [19], as shown in the following equation.

$$I^c(x, y) = J^c(x, y)e^{-\beta d(x, y)} + A^c(1 - e^{-\beta d(x, y)}) \quad (5)$$

where $I^c(x, y)$ represents the intensity of the elements in the image obtained at the position (x, y) , J^c is the light reflected by the objects without haze, A^c is the ambient

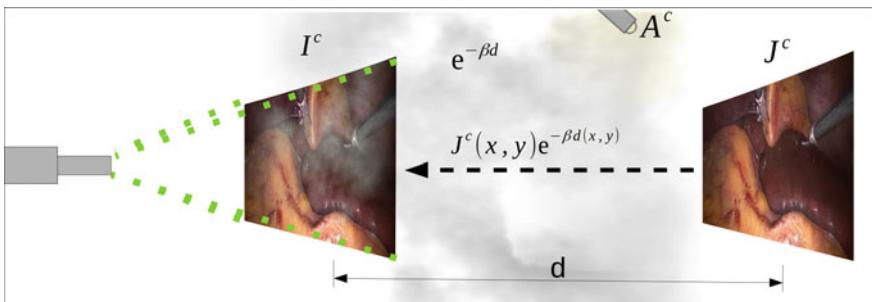


Fig. 2 Hazy image formation, I^c represents the image obtained, J^c is the light reflected or generated by objects in the direction of the observer, A^c is the ambient light and $e^{-\beta d}$ is the transmission map

light, and c represents each RGB color. Also, $e^{-\beta d(x,y)}$ is the transmission map, where β is the scattering coefficient, and d is the scene depth.

Given Eq. 5, it is observed that if there are no light scattering elements $\beta \approx 0$, then $I \approx J$, however in the presence of vapors, β becomes a very significant value. It is observed that the direct attenuation $J^c(x, y)e^{-\beta d(x,y)}$ increases with decreasing depth, and the effect of ambient light $A^c(1 - e^{-\beta d(x,y)})$ decreases with decreasing scene depth, thus direct attenuation is the most relevant effect of smoke degradation in laparoscopic images.

4.1.1 Smoke Simulation

In Sect. 5.1.1, the dataset required the inclusion of a realistic smoke simulation, which is crucial for generalizing the smoke behavior in real images. To simulate the smoke, we used the atmospheric dispersion model, given in Eq. 6. Furthermore, we generated transmission maps ($m(x, y)$) using the Python Cloud code presented in [20], to generate 255×255 px cloud masks. Therefore $m(x, y)$ replaces $e^{-\beta d(x,y)}$ in Eq. 5 to generate realistic images. An example of the generation of these images is shown in Fig. 3.

$$I^c(x, y) = J^c(x, y)m(x, y) + A^c(1 - m(x, y)) \quad (6)$$

where $m(x, y)$ represents the smoke_{intensity}, ($0 \leq \text{smoke}_{\text{intensity}} \leq 1$).

4.2 Dark Channel Prior (DCP)

The presence of smoke in laparoscopic surgery images can be effectively detected by analyzing the dark channel. By examining the dark pixels in the red (R), green (G), and blue (B) channels, it is possible to identify the potential presence of smoke. In smoke-free or haze-free images, these dark pixels have intensity values near zero

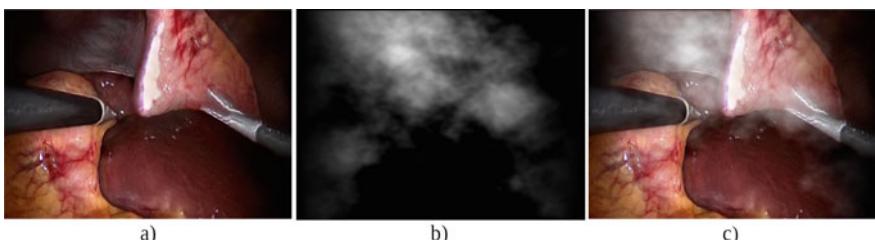


Fig. 3 Example of image generation with synthetic smoke. **a** Original image. **b** Transmission map. **c** Synthetic smoke image

in at least one color channel throughout most regions [21, 22]. This relationship can be expressed mathematically in the following Eq. 7.

$$I^{\text{dark}}(x, y) = \min_{c \in \{R, G, B\}} \left(\min_{z \in \Omega(x, y)} I^c(z) \right), \quad (7)$$

where (x, y) represent the coordinates of each pixel in the image, $\Omega(x, y)$ is a kernel (usually squared) centered at the (x, y) position, $I^c(z)$ are the elements of the smoked laparoscopic image I at the positions $z \in \Omega(x, y)$, and c represents each *RGB*. This means that $I^{\text{dark}} \rightarrow 0$ in smoke-free images. This statistical fact is called *dark channel prior*. In this case, the dark-channel can be used as a mask to provide additional information to the network to improve its performance for this task.

In laparoscopic images, this relation still applies, as tissues, organs and fat are usually opaque in at least one color channel. A comparison between the dark channel of a laparoscopic image with and without smoke can be seen in Fig. 4.

4.2.1 Dark Channel Refinement

The dark channel I^{dark} is not exactly aligned with the image I . To reduce this spatial inaccuracy, it is mandatory to integrate a refinement stage, a proposed methods for this can be seen in [22]. In order to obtain a spatially accurate map of the dark channel shown in Eq. (5), we apply a *guided filter* to I^{dark} , which is an edge-preserving smoothing filter based on a linear model in a region [23].

$$I_{\text{ref}}^{\text{dark}}(z) = a(x, y)I(z) + b(x, y), \forall z \in \Omega(x, y), \quad (8)$$

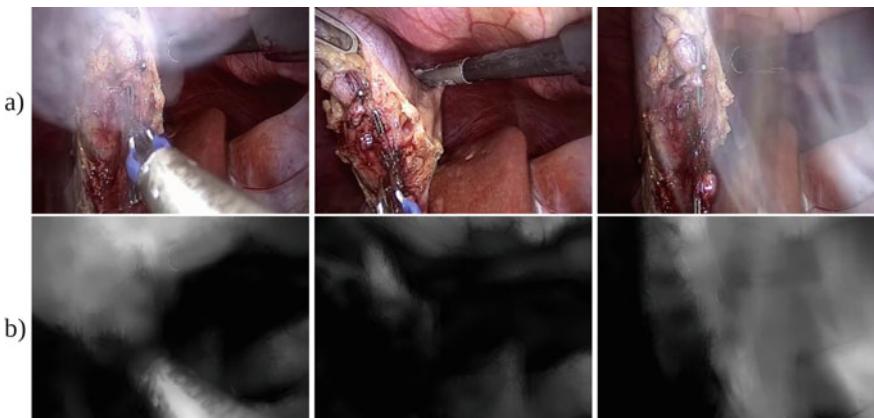


Fig. 4 Comparison of the dark channel with different smoke density. **a** Original image. **b** Dark channel mask

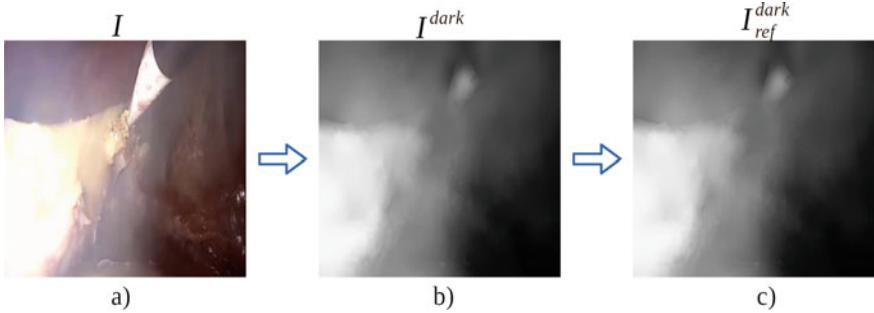


Fig. 5 Process of estimation and refinement of the dark channel in the smoked laparoscopic image. **a** Original image. **b** Dark channel. **c** Refined dark channel

where $I_{\text{ref}}^{\text{dark}}$ is the filtering output dark channel, I is the guidance image, and z is the position of a pixel in the local squared window Ω , of size $s \times s$ and centered at (x, y) . The $a(x, y)$ and $b(x, y)$ parameters from linear model (8) are defined as

$$a(x, y) = \frac{\frac{1}{|\Omega|} \sum_{(z) \in \Omega(x, y)} I(z) I^{\text{dark}}(z) - \mu(x, y) \overline{I^{\text{dark}}(x, y)}}{\sigma(x, y)^2 + \epsilon}, \quad (9a)$$

$$b(x, y) = \overline{I^{\text{dark}}(x, y)} - a(x, y) \mu(x, y), \quad (9b)$$

where $\mu(x, y)$ and $\sigma(x, y)$ are the mean and variance of I in $\Omega(x, y)$, $\overline{I^{\text{dark}}(x, y)}$ is the mean of $I^{\text{dark}}(x, y)$ in Ω_k , and ϵ is parameter that regulates the smoothness degree. The computed and refined dark channel $I_{\text{ref}}^{\text{dark}}(z)$ is stacked into the smoked laparoscopic image I , as shown in Fig. 5.

4.3 Segmentation

As noted in the previous section and in the earlier article [24], the refined dark channel serves as a valuable reference for the neural network, providing a guide for smoke removal applications by highlighting specific areas for attention. Nevertheless, certain tissues and instruments possess optical properties that make them visible in the dark channel. However, their discernible physical structure and texture clearly indicate that they are not smoke, as shown in Fig. 7. Therefore, image component segmentation is beneficial because it provides the network with a reliable reference regarding the presence of tissues or tools in a given area.

Figure 6 shows a complete segmentation of multiple images extracted from the dataset [25]. It provides a clear visual representation of the identification and differentiation of organic tissues, fat and flesh, as well as the various components of

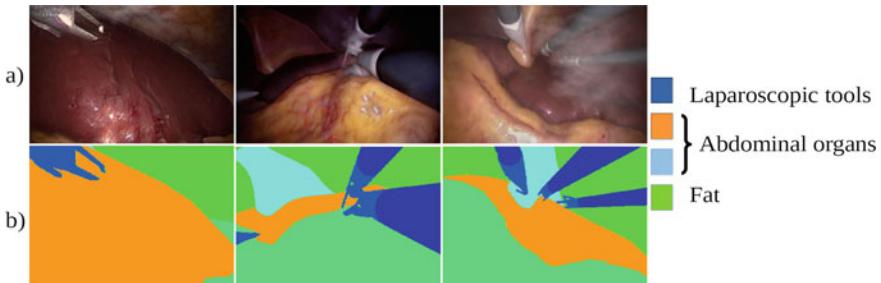


Fig. 6 Semantic segmentation and corresponding colors. **a** Original image. **b** Semantic segmentation

the laparoscopic tools. The segmentation process allows accurate delineation and recognition of these elements within the images, providing valuable data on their distinguishing characteristics and spatial arrangement.

In Fig. 7, the dark channel not only reveals regions of smoke, but also highlights laparoscopic tools and lighter tissues, but these areas are well identified by the segmentation map, with the tools represented in dark blue and the lighter tissues in light blue. Using these two guides, the network gains the necessary information to efficiently remove smoke from specific areas of the image.

5 GAN Powered by Dark-Channel and Segmentation

The neural network designed for this particular task is a fusion of two Conditional Generative Adversarial Networks (C-GANs). This architecture is specifically tailored to address the challenges posed by laparoscopic images contaminated with smoke. In the initial stage of the network, a two-fold input consisting of the laparoscopic image with smoke (R,G,B) and its corresponding dark-channel map is fed into the system, the first C-GAN processes this input to generate a segmentation map. This segmentation map plays a vital role in distinguishing and categorizing the distinct components within the laparoscopic images. By effectively identifying and separating various structures and elements present in the images, the segmentation map lays the groundwork for subsequent analysis and processing.

In the next stage, the input data, comprising the original laparoscopic image and the calculated dark-channel map, is merged with the output from the previous step: the segmentation map. This fusion results in a new, augmented image with an expanded dimensionality of five channels. This image, which now contains vital information about both the original image, the dark-channel map and the corresponding segmentation map, serves as the input for the final stage of the network, this concluding phase of the network is specifically trained to remove smoke from the laparoscopic images. Leveraging the multi-channel information provided by the augmented input

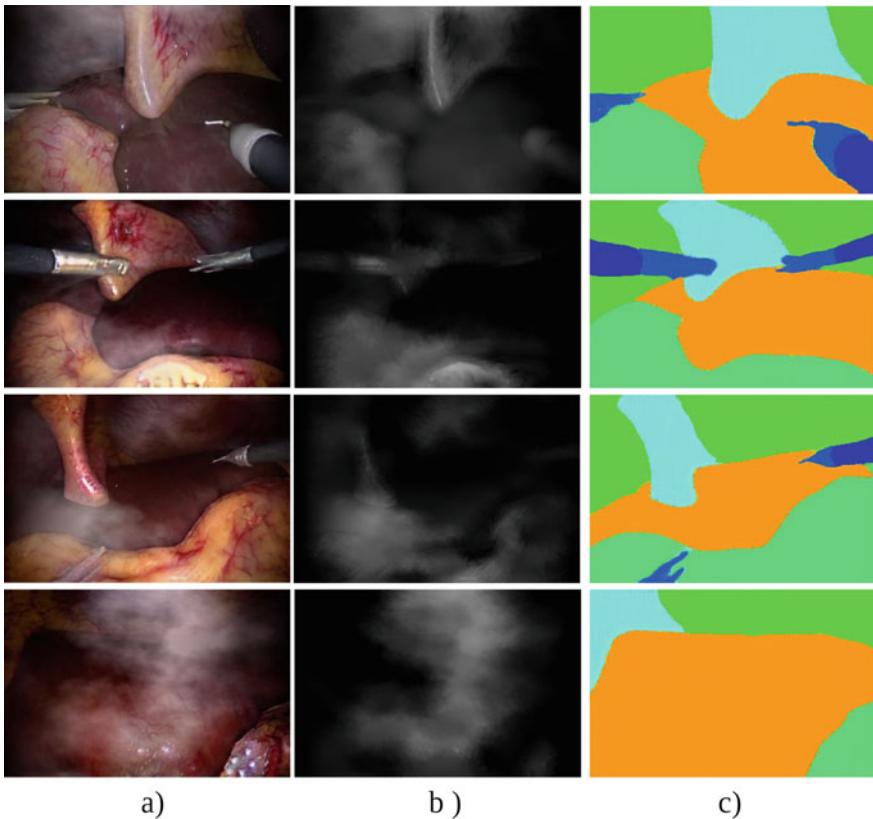


Fig. 7 Comparison of dark channel and semantic segmentation with different tissues and tools. **a** Original image **b** Refined dark channel. **c** Semantic segmentation

image, to effectively eliminate smoke artifacts and enhance the visual quality of the images. By combining the power of two C-GANs and leveraging the distinctive contributions of the dark-channel map and the segmentation map, this neural network architecture demonstrates its capability to address the challenges associated with smoke-contaminated laparoscopic images. Its multi-stage design and the incorporation of various data sources allow for improved accuracy and effectiveness in smoke removal, ultimately contributing to enhanced image quality and visibility in the laparoscopic procedures.

The proposed architecture can be visually represented as shown in Fig. 8, additionally can be formally described as follows:

1. Segmentation map generation:

$$S_{\text{seg}} = C_{\text{GAN}_1}(I_l \text{ with smoke}, M_d) \quad (10)$$

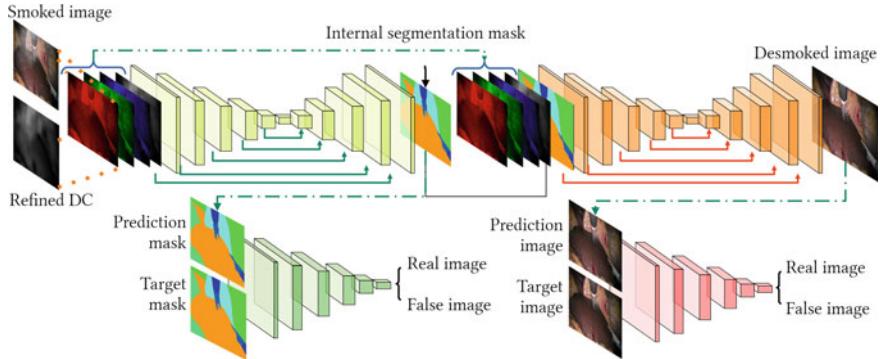


Fig. 8 Architecture of the proposed network

The network C_{GAN_1} takes the input laparoscopic image with smoke and the corresponding dark-channel map as input and produces the segmentation map.

2. Image augmentation:

$$I_{\text{aug}} = \text{Concatenate}(I_l, M_d, S_{\text{seg}}) \quad (11)$$

The augmented image I_{aug} is formed by concatenating the original laparoscopic image, the dark-channel map, and the segmentation map.

3. Post-processing:

$$I_{\text{pp}} = C_{GAN_2}(I_{\text{aug}}) \quad (12)$$

The network C_{GAN_2} takes the augmented image as input and performs post-processing to generate the final output image I_{pp} .

where

- I_l : Input laparoscopic image with smoke
- M_d : Dark-channel map computed from I_l
- S_{seg} : Output segmentation map
- I_{aug} : Augmented image
- I_{pp} : Post-processed image
- C_{GAN_1} : Conditional generative adversarial network for segmentation
- C_{GAN_2} : Conditional generative adversarial network for post-processing.

The detailed description of the internal layers is shown in Tables 1 and 2.

Table 1 Architecture of the generator G . Capital letter C denotes (Convolution \rightarrow BatchNorm \rightarrow Leaky ReLU); CT denotes (Deconvolution \rightarrow BatchNorm \rightarrow ReLU), all kernels are of 4 and strides of 2 \rightarrow ReLU \rightarrow Dropout rate 50%; and CT denotes (Deconvolution \rightarrow BatchNorm \rightarrow ReLU), all kernels are of 4 and strides of 2

	C_{GAN_1}			C_{GAN_2}		
Input : I_1 with smoke, M_d						
Output: \rightarrow Input C_{GAN_2}	Layer	Conv	Definition	Size	Skip connection	Layer
1	64	C	(128, 128, 64)	$\rightarrow 16$	18	64
2	128	C	(64, 64, 128)	$\rightarrow 15$	19	128
3	256	C	(32, 32, 256)	$\rightarrow 14$	20	256
4	512	C	(16, 16, 512)	$\rightarrow 13$	21	512
5	512	C	(8, 8, 512)	$\rightarrow 12$	22	512
6	512	C	(4, 4, 512)	$\rightarrow 11$	23	512
7	512	C	(2, 2, 512)	$\rightarrow 10$	24	512
8	512	C	(1, 1, 512)	n/a	25	512
9	512	CTD	(1, 1, 1024)	n/a	26	512
10	1024	CTD	(2, 2, 1024)	$7 \leftarrow$	27	1024
11	1024	CTD	(4, 4, 1024)	$6 \leftarrow$	28	1024
12	1024	CT	(8, 8, 1024)	$5 \leftarrow$	29	1024
13	1024	CT	(16, 16, 1024)	$4 \leftarrow$	30	1024
14	512	CT	(32, 32, 512)	$3 \leftarrow$	31	512
15	256	CT	(64, 64, 256)	$2 \leftarrow$	32	256
16	128	CT	(128, 128, 128)	$1 \leftarrow$	33	128
17	n/a	tanh	(256, 256, 3)	$\rightarrow input C_{GAN_2}$	34	n/a
				Input: Concat($Input C_{GAN_1}$, $Output C_{GAN_1}$)		
				Output: Final Output = I_{pp}		
					Definition	Size
					C	(128, 128, 64)
					C	(64, 64, 128)
					C	(32, 32, 256)
					C	(16, 16, 512)
					C	(8, 8, 512)
					C	(4, 4, 512)
					C	(2, 2, 512)
					C	(1, 1, 512)
					CTD	(1, 1, 1024)
					CTD	(2, 2, 1024)
					CTD	(4, 4, 1024)
					CT	(8, 8, 1024)
					CT	(16, 16, 1024)
					CT	(32, 32, 512)
					CT	(64, 64, 256)
					CT	(128, 128, 128)
				tanh	n/a	(256, 256, 3)
						n/a

Table 2 The architecture of the employed discriminator D for C_{GAN_1} and C_{GAN_2}

Layer	Conv	kernel	Strides	Definition	Size
1	64	4	2	(Conv ->BatchNorm ->Leaky ReLU)	(128, 128, 64)
2	128	4	2	(Conv ->BatchNorm ->Leaky ReLU)	(64, 64, 128)
3	256	4	2	(Conv ->BatchNorm ->Leaky ReLU)	(32, 32, 256)
4	0	0	0	(ZeroPadding2D)	(34, 34, 256)
5	512	4	1	(Conv)	(31, 31, 512)
6	0	0	0	(BatchNorm ->Leaky ReLU ->ZeroPadding)	(33, 33, 512)
7	1	4	1	(Conv)	(30, 30, 1)

5.1 Training

5.1.1 Dataset

The dataset used for segmentation training was obtained from [25]. It consists of 100,000 synthetic images generated in a 3D environment, resembling scenes of laparoscopic hepatic surgery. These images were processed by a GAN to transform their appearance into realistic laparoscopy images.

The training data for the dehazing network was sourced from the Cholec80 dataset, which consists of 80 videos of cholecystectomy surgeries performed by 13 surgeons. In order to prepare the dataset for training, from a subset of 50 videos within the dataset, a total of 20,000 representative images without any smoke or hazy artifacts were extracted. To simulate real-world conditions and enhance the network's ability to handle hazy environments, synthetic smoke was introduced to these images see Sect. 4.1.1. This process involved randomly adding smoke to the original images, resulting in pairs of input and output images. The input images were representative of the smoke-affected scenes, while the output images retained the original, smoke-free appearance. These image pairs were used to train the complete dehazing network shown in Fig. 8.

5.1.2 Hyperparameters

The network architectures of the proposed network are shown in Fig. 8, Both are trained for 120 epochs with 60,000 pairs of images of 256×256 pixels for the input and output, We used a batch size of 250, a constant learning rate of 0.0002. Weights were initialized using a Gaussian distribution with zero mean and a standard deviation of 0.02. the ADAM optimizer was used.

5.1.3 Setup

The experiments were performed on a workstation equipped with an AMD Ryzen 5 5600G CPU boasting 64 GB of RAM and an Nvidia RTX A6000 GPU with 48GB of VRAM. The operating system employed was Ubuntu 20.04, and the models were built using Pytorch.

6 Metrics

Accurate evaluation of an image processing model's performance is crucial to understand its effectiveness and compare it with other approaches. In this study, several non-referenced image quality metrics were employed to obtain a comprehensive overview of the performance of the proposed model and the comparative models. These metrics include BRISQUE (Blind/Referenceless Image Spatial Quality Evaluator) [26], NIQE (Natural Image Quality Evaluator) [27], PIQE (Perceptual Image Quality Evaluator) [28], and FADE (Full-Reference Assessment of Digital Video Quality) [29].

BRISQUE is a blind image quality assessment (IQA) metric that estimates the quality of an image without requiring a reference image. BRISQUE is based on the analysis of statistical properties of an image in the spatial domain. It computes a quality score that reflects the level of distortion and artifacts present in the image. Higher BRISQUE scores indicate lower image quality, while lower scores indicate higher image quality.

NIQE is another blind IQA metric that focuses on assessing the quality of natural images. NIQE measures the amount of natural image information present in an image by comparing the statistical properties of the image with a model of natural image statistics. NIQE provides a quality score that indicates the perceived quality of the image, with higher scores indicating lower quality and lower scores indicating higher quality.

PIQE is a perceptual IQA metric that focuses on the human visual system's perception of image quality. PIQE uses a representative set of images to assess the perceived quality of an image based on visual and perceptual features.

FADE is a full-reference IQA metric that requires a reference image to calculate the quality score. FADE is based on the analysis of statistical properties in the DCT (Discrete Cosine Transform) domain of video sequences.

The inclusion of these metrics provides a more comprehensive and detailed evaluation of the performance of the proposed model, covering aspects of spatial quality, perceptual quality, and video quality. Their utilization allows for a more accurate comparison with other approaches and a more realistic evaluation in the context of real laparoscopy images.

7 Results and Discussion

In order to evaluate the performance and highlight the contributions of the proposed method, it is important to compare it with other state-of-the-art methods. This comparative analysis provides a comprehensive understanding of the strengths and weaknesses of the new approach and demonstrates its superiority in terms of results.

The methods to be compared include: CycleGAN [30]: This method proposes a cycle-consistent adversarial network-based unpaired image-to-image translation approach. It has been widely applied in various image processing applications and has demonstrated its effectiveness in unpaired image translation tasks. FastCUT [31]: This research focuses on contrastive learning for unpaired image-to-image translation. It introduces improvements in generating realistic images and aligning latent features in unpaired image translation. Colores et al. [24]: This method employs an embedded dark channel-guided image-to-image translation to remove smoke in laparoscopic images. It shows promising results in smoke removal, but its effectiveness and generalization need to be analyzed and compared. Cycle-Dehaze [32]: This method presents an enhanced version of the cycle GAN for single image dehazing. It has been utilized to improve visibility in hazy images and may be relevant for comparison in terms of improvements in the clarity of laparoscopic images. Desmoke-Lap [33]: This approach focuses on enhanced unpaired image-to-image translation for smoke removal in laparoscopic surgery. It provides improvements in visual quality and more accurate results in smoke removal, making it a relevant method for comparison.

As mentioned earlier, four non-referenced metrics were utilized: BRISQUE (Fig. 9), FADE (Fig. 10), NIQE (Fig. 11), and PIQE (Fig. 12).

7.1 Quantitative Analysis

Our proposed architecture demonstrated consistent and impressive performance across all evaluated metrics, consistently ranking among the top two positions.

Furthermore, it is worth highlighting that in three out of the four cases, our proposed model outperformed the baseline model of Colores et al. [24], which served as the foundation for our current research. This showcases the advancements and improvements achieved through our proposed architecture, surpassing the performance of the previous state-of-the-art method. The results suggest that our approach not only addresses the limitations of the baseline model but also offers superior results in terms of smoke removal in laparoscopic surgery images.

These findings validate the effectiveness and significance of our proposed architecture, reinforcing its potential for practical applications in the field of laparoscopic image processing. The consistent top-ranking performance and superiority over the baseline model demonstrate the contributions and advancements of our research in smoke removal and image enhancement.

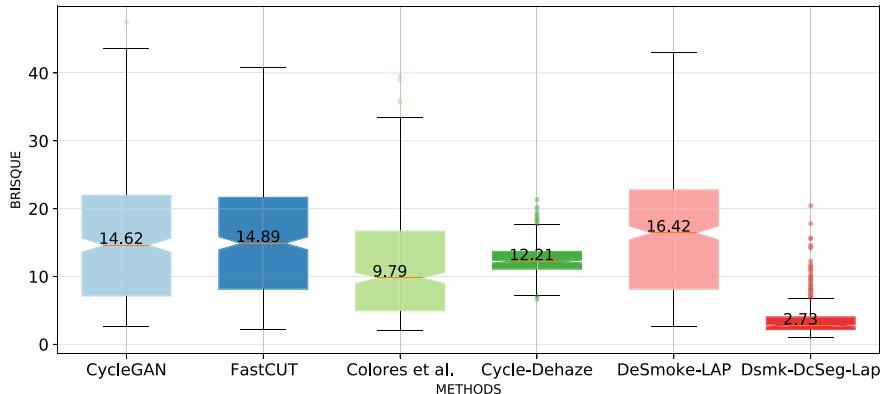


Fig. 9 A comparative analysis of the performance between state-of-the-art methods and the proposed methodology, according to BRISQUE index

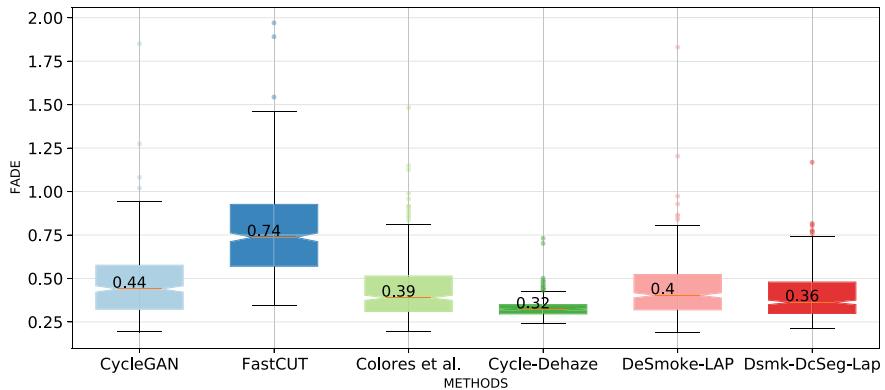


Fig. 10 A comparative analysis of the performance between state-of-the-art methods and the proposed methodology, according to FADE index

7.2 Qualitative Analysis

Furthermore, as depicted in Fig. 13, a qualitative comparison of the processed images by various algorithms is presented. Notably, our method effectively preserves the original tonalities of the image while solely addressing the affected areas caused by smoke. In contrast, algorithms such as CycleGAN, FasCUT, Cycle-Dehaze, and DSmoke-LAP exhibit alterations in the tonalities of the images. This highlights the advantage of our approach in maintaining the fidelity of the original image content.

Moreover, our method possesses the desirable characteristic of avoiding modifications to the image if the input is unaffected by smoke. This ensures that the algorithm does not introduce unnecessary changes to the image when smoke is not present, preserving the integrity of the original image composition.

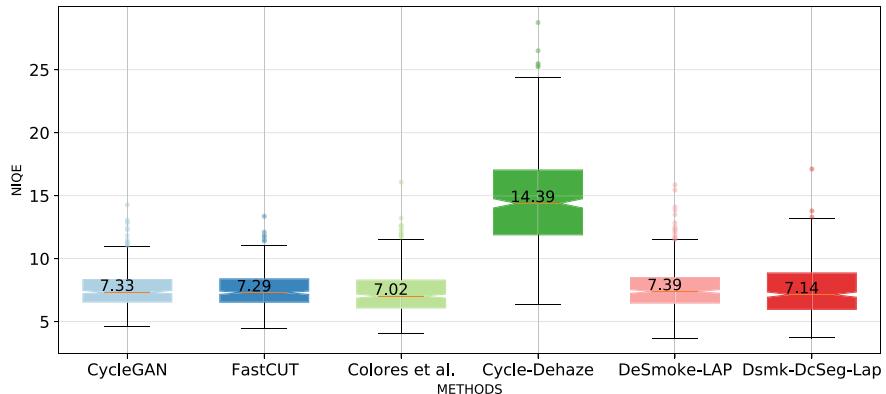


Fig. 11 A comparative analysis of the performance between state-of-the-art methods and the proposed methodology, according to NIQE index

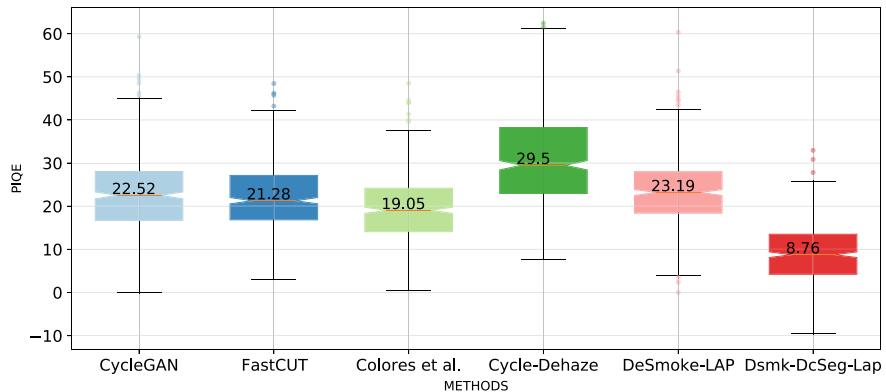


Fig. 12 A comparative analysis of the performance between state-of-the-art methods and the proposed methodology, according to PIQE index

It is important to note that the test dataset utilized in these experiments was extracted from the Desmoke-LAP dataset [33]. It is crucial to emphasize that our network was not exposed to this dataset during any stage of the training process. This ensures that the evaluation is performed on unseen data, enhancing the reliability and generalizability of our results.

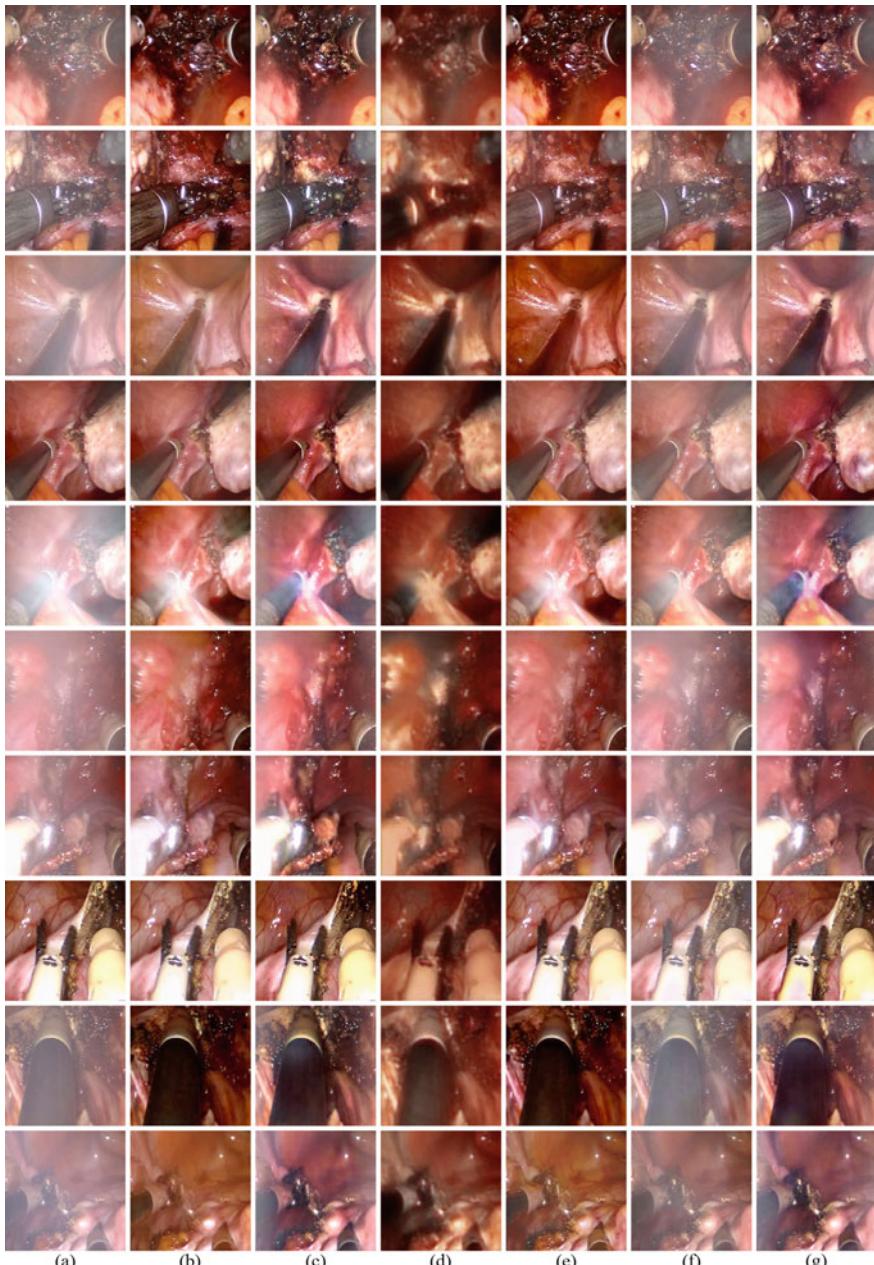


Fig. 13 Comparison with five state-of-the-art methods in real laparoscopic images. **a** Input. **b** CycleGAN. **c** FastCUT. **d** Colores et al. **e** Cycle-Dehaze **f** DeSmoke-LAP. **g** Our proposed method (DsmK-DcSeg-Lap)

8 Conclusion

At the beginning of this chapter, it was explained that during laparoscopic surgeries, the surgeon's visibility can be obstructed due to smoke generated during tissue cauterization. These factors can lead to prolonged surgery duration and potential surgical errors. To address these issues, this study proposed a solution involving the utilization of two interconnected conditional antagonistic generative networks (C-GANs) and the dark channel embedded in the input image. This network aims to generalize the smoke behavior and eliminate it. By leveraging the guidance provided by the dark channel and the internally generated semantic segmentation.

The inclusion of semantic segmentation and the dark channel as guiding factors significantly contributes to a centralized smoke removal approach, particularly targeting the affected areas. The effectiveness of this method was demonstrated through evaluations conducted on both synthetic and real images. According to the metrics used, the proposed network consistently achieved top rankings in two of the evaluated metrics and secured the second position in the remaining ones, showcasing its superiority over alternative algorithms, which exhibited more variable outcomes. In the future, work could be done on developing improved loss functions for this network and experimenting with connecting internal layers of the network.

Acknowledgements Hugo Moreno would like to express his sincere gratitude to CONAHCYT (National Council for Humanity, Science and Technology) for their financial support of his doctoral studies.

References

1. Rizzo, G., Cuccurullo, D., Iacobelli, G., & et al. (2019). Smoke evacuation systems in laparoscopic surgery: A systematic review and meta-analysis. *Surgical Endoscopy*, 33(11), 4507–4516.
2. Bhatia, S., Singh, N., Kaur, S., & et al. (2020). Smoke management in laparoscopy: a comprehensive review. *Journal of Minimal Access Surgery*, 16(1), 18.
3. Fernandez-Ruiz, J., Perez-de Luque, A., Perez-Castellanos, J. L., & et al. (2021). Novel smoke evacuation systems for laparoscopy: A systematic review. *Journal of Laparoendoscopic and Advanced Surgical Techniques*, 31(1), 1–10.
4. Smith, J., & Johnson, A. (2010). Smoke removal from images using adaptive filtering. *IEEE Transactions on Image Processing*, 19(5), 1125–1133.
5. Johnson, R., & Lee, H. (2012). Image dehazing using non-linear techniques. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1674–1681). IEEE.
6. Brown, M. S. (2015). Multi-spectral deep learning for smoke detection in video. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2793–2801). IEEE.
7. Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798–1828.
8. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.

9. Goodfellow, I., & et al. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2672–2680.
10. Smith, J., & Johnson, E. (2018). Physics-based smoke removal using deep neural networks. *IEEE Transactions on Image Processing*, 27(3), 1340–1353.
11. Lee, C., & et al. (2020). Image dehazing using domain adaptation. In *Proceedings of the IEEE international conference on computer vision (ICCV)* (pp. 1122–1130). IEEE.
12. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.
13. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 1097–1105.
14. Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 807–814).
15. Boureau, Y. -L., Ponce, J., & LeCun, Y. (2010). Theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 111–118).
16. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 234–241). Springer.
17. Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint [arXiv:1411.1784](https://arxiv.org/abs/1411.1784)
18. Isola, P., Zhu, J. -Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125–1134).
19. Lee, S., Yun, S., Nam, J.-H., Won, C. S., & Jung, S.-W. (2016). A review on dark channel prior based image dehazing algorithms. *EURASIP Journal on Image and Video Processing*, 2016, 1–23.
20. Zhang, N., Zhang, L., & Cheng, Z. (2017). Towards simulating foggy and hazy images and evaluating their authenticity. In *International conference on neural information processing* (pp. 405–415). Springer.
21. He, K., Sun, J., & Tang, X. (2010). Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12), 2341–2353.
22. Chandra, A., Singh, A., Kumar, R., & Dey, N. (2018). Dehazing of aerial images by dark channel and gamma correction. In *2018 3rd International conference and workshops on recent advances and innovations in engineering (ICRAIE)* (pp. 1–7). IEEE.
23. He, K., Sun, J., & Tang, X. (2010). Guided image filtering. In K. Daniilidis, P. Maragos, & N. Paragios (Eds.), *Computer vision—ECCV 2010* (pp. 1–14). Springer.
24. Salazar-Colores, S., Jiménez, H. M., Ortiz-Echeverri, C. J., & Flores, G. (2020). Desmoking laparoscopy surgery images using an image-to-image translation guided by an embedded dark channel. *IEEE Access*, 8, 208898–208909.
25. Pfeiffer, M., Funke, I., Robu, M. R., Bodenstedt, S., Strenger, L., Engelhardt, S., Roß, T., Clarkson, M. J., Gurusamy, K., Davidson, B. R., & et al. (2019). Generating large labeled data sets for laparoscopic image processing tasks using unpaired image-to-image translation. In *Proceedings of the 22nd international conference on medical image computing and computer assisted intervention—MICCAI 2019* (pp. 119–127), Shenzhen, China, October 13–17, 2019, Part V 22. Springer.
26. Mittal, A., Moorthy, A. K., & Bovik, A. C. (2012). No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12), 4695–4708.
27. Mittal, A., Moorthy, A. K., & Bovik, A. C. (2012). Making a “completely blind” image quality analyzer. *IEEE Signal Processing Letters*, 20(3), 209–212.
28. Moorthy, A. K., & Bovik, A. C. (2010). Perceptual image quality assessment using a representative set of images. *IEEE Transactions on Image Processing*, 19(8), 2368–2382.
29. Saad, M. H., Bovik, A. C., & Charrier, C. (2012). Blind image quality assessment: A natural scene statistics approach in the DCT domain. In *2012 19th IEEE International conference on image processing* (pp. 717–720). IEEE.

30. Zhu, J. -Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223–2232).
31. Park, T., Efros, A. A., Zhang, R., & Zhu, J. -Y. (2020). Contrastive learning for unpaired image-to-image translation. In *Proceedings of the 16th European conference on Computer vision–ECCV 2020* (pp. 319–345), Glasgow, UK, August 23–28, 2020, Part IX 16. Springer.
32. Engin, D., Genc, A., & Kemal Ekenel, H. (2018). Cycle-dehaze: Enhanced cylegan for single image dehazing. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 825–833).
33. Pan, Y., Bano, S., Vasconcelos, F., Park, H., Jeong, T. T., & Stoyanov, D. (2022). DeSmoke-LAP: improved unpaired image-to-image translation for DeSmoking in laparoscopic surgery. *International Journal of Computer Assisted Radiology and Surgery*, 17(5), 885–893.

Generative AI Use in the Construction Industry



Gozde Basak Ozturk and Fatih Soygazi

Abstract This chapter presents a comprehensive exploration of Generative Artificial Intelligence (AI) and its applications in the Architecture, Engineering, Construction, and Facility Management (AEC-FM) industry. It begins with a general overview of Generative AI, highlighting its capacity to create novel content. The subsequent section delves into the technological requirements for implementing Generative AI in the AEC-FM industry, including Internet of Things (IoT), Distributed Ledger Technology, Computing, Deep Learning, Natural Language Processing (NLP), Knowledge Graph, Computer Vision, and Immersive Technologies. The chapter then proceeds to discuss the wide-ranging applications of Generative AI in the AEC-FM industry, showcasing its potential in enhancing design processes, predictive modeling, operations, and maintenance. Furthermore, the role of Building Information Modeling (BIM) as a facilitator for Generative AI in the AEC-FM industry is explored. Identifying gaps and trends in Generative AI research, the chapter presents a keyword co-occurrence analysis specific to the AEC-FM industry, covering life-cycle management, predictive modeling, machine learning implementation in design, design optimization, constraint-based generative design, and design techniques. Finally, the chapter concludes by examining the future prospects for Generative AI in AEC-FM activities, offering insights into its potential transformative impact on the industry.

Keywords Construction industry · BIM · Digital twins · Generative AI · Generative design

G. B. Ozturk (✉)

Faculty of Engineering, Department of Civil Engineering, Aydin Adnan Menderes University,
Aydin 09100, Turkey

e-mail: gbozturk@adu.edu.tr

F. Soygazi

Faculty of Engineering, Department of Computer Engineering, Aydin Adnan Menderes
University, Aydin 09100, Turkey

e-mail: fatih.soygazi@adu.edu.tr

1 A General Overview for Generative AI

Generative AI is a type of artificial intelligence technology that can produce various types of content, including text, imagery, audio and synthetic data. It is capable of generating seemingly new, realistic content—such as text, images, or audio—from the training data which is acquired from the related domain. This data whether can be gathered from a digital storage, or from non-digital (paper-based) information. The accumulated big data is used for multi-alternative requisition in generating creative outputs. Thereby, high-quality text, graphics, and videos are produced rapidly.

The initial steps for Generative AI started with the alternatives were produced via rule-based methods and heuristic methods (genetic algorithms, beam search, simulated annealing, ant colony optimization, bee colony optimization) [1]. In the last decade, the accelerated increase in big data, the lower cost of processing power, the emergence of the novel processing power hardware, traditional artificial neural networks have been started to being used in a more layered manner called as deep learning. Afterwards, increasing needs for the complex output expectations has led to a more focused deep learning structure. As a result, Attention Mechanism attracted scholars in all fields. In Generative AI, the attention mechanism has enabled models to learn and produce better outputs.

Previous Generative AI models faced difficulties in handling long-range dependencies, especially in complex data types like language or images. For example, a language model struggled when a word at the beginning of a sentence depended on a word at the end. Similarly, in image generation, a model encountered issues when a pixel relied on its previous pixels. For instance, the method can generate alternative for structural design of a building with taking the architectural style into account at the same time. The attention mechanism has helped Generative AI models better addresses these long-range dependencies. In transformer models, the attention mechanism uses weights to determine the relationships between each generated element and other elements. These weights indicate how the model will interact with relevant elements during the generation process. Thereby, the chaotic parametric decisions across domain alternatives can be generated for various purposes. For example, the scheduling of the building can be controlled with testing the sustainability performance simultaneously during the execution of the building. It is possible with Generative AI to detect the relationship of a data-oriented feature (word in NLP, pixels in image processing, columns in tabular data, etc.) in an intensive manner with a contextual perspective. For example, a sensor data from concrete mixer can inform the system for an accident, so that the concrete pouring activity can be regenerated by Generative AI to optimize scheduling. The Transformer Models are used for various purposes with different types via facilitating deep learning models by the big companies such as OpenAI, Microsoft, Google etc. Generative AI models are continuously improved through iterative refinement. Researchers and developers collect feedback on the generated outputs and fine-tune the models further to enhance their accuracy, coherence, and overall quality.

2 Technological Requirements for Generative AI Application in the AEC-FM Industry

While traditional methods have their merits, Generative AI in the AEC-FM industry hold the potential to revolutionize the design, construction, and management processes, leading to more efficient, sustainable, and innovative built environments. Continued advancements in AI research, data availability, and computational power will likely contribute to the realization of these prospects in the coming years. Generative AI purposes, use cases in the AEC-FM industry, and enhancing technologies are listed in Table 1.

Table 1 Generative AI, purposes, use cases, and enhancing technologies

Purpose of generative AI	Use case in the AEC-FM industry	Enhancing technologies
Content generation	Design exploration, BIM, generate 3D models from 2D drawings	Computing, DLT, deep learning, NLP, KG, computer vision, immersive technologies
Data augmentation	BIM, generate 3D models from 2D drawings, spatial relationships, automatic BIM workflow generation, data extraction from point cloud models,	IoT, computing, deep learning, NLP, KG, computer vision
Personalization	Site team management, interior design, landscape design, in-building routing	Computing, DLT, deep learning, KG, computer vision, immersive technologies
Simulation and modeling	Design optimization, generative structural analysis, material usage, layout orientation alternatives, energy efficient configuration, structural stability, cost effectiveness, sustainable design, sequencing, scheduling, resource allocation	Computing, deep learning, computer vision, immersive technologies
Creative assistance	Predictive maintenance, logistics planning, data-driven decision making	Computing, IoT, deep learning, NLP, KG, computer vision, immersive technologies

2.1 Internet of Things (IoT)

The internet of things (IoT) concept originated in the 1980s and gained momentum with the idea of connecting appliances [2]. The integration of radio-frequency identification (RFID) and IoT formed the foundation of this concept. Research in IoT has seen significant growth, particularly in the architecture, engineering, construction, and facility management (AEC-FM) industry since 2015 [3–7]. IoT involves the communication, computation, and coordination among machines and objects via the internet, with data being collected, processed, and stored in cloud-based servers [8–11]. IoT platforms consist of layers of sensors, processors, gateways, and applications, enabling data collection, local storage, and cloud-based management [12–16]. Various communication technologies, such as RFID, wireless sensor networks (WSN), near field communication (NFC), ZigBee, Bluetooth, Wi-Fi, and cellular networks, are used for IoT implementation. IoT finds applications in supply chain management, healthcare, smart home, robotics, transportation, manufacturing, agriculture, retail, and other areas [17–24]. The benefits of using IoT technology eases and simplifies to automate, achieve, and control tasks [5, 25]. However, security and privacy are the main considerations in IoT implementation [26, 27]. The IoT elements needed for a functional implementation are namely identification, communication, sensing, services, computation, and semantics.

Dave et al. [28] address the need to systematically integrate IoT technologies and the built environment through open standards and systems. The proposed platform is called Otaniemi3D that integrates building data and his IoT sensors into his web-based system across the campus. The platform uses open messaging standards and building information models to provide information on energy consumption, occupancy and user comfort. The research describes the design criteria, system architecture, workflow, and presents a proof of concept with possible use cases.

Ozturk [29] discusses the possibility of combining Building Information Modeling (BIM) and Internet of Things (IoT) technologies to create a digital twins of the built environment. There is a high potential of Digital Twins for improving AEC-FM activities. However, AI integration to the platform is vital for enhancing full potential of the digital twins platform by increasing the maturity level. This might be realized through adoption of the latest AI approaches such as Generative AI.

2.2 Distributed Ledger Technology

Digital systems that provide decentralized databases are called Distributed Ledger Technology (DLT). These digital systems eliminate the central authority hence it spreads digital data across multiple places at the same time. DLT controls against manipulation of the network and allows for record-keeping of all information in the system safe and secure with cryptographic algorithms. Blockchain is the most prominent type of DLTs that could be integrated with Generative AI for increasing Digital

Twins Maturity. Blockchain technology is used for transactions such as cryptocurrencies, goods, assets, money, votes, and digital identities. There are several use cases in the AEC-FM industry. BIM model ownership is in the focus of the researchers. Blockchain offers solutions for BIM implementation challenges which in turn will be a base for Digital Twins applications. The maturity level of the Digital Twins is directly related to the system's ability to serve secure and reliable knowledge which is generated from robust data. Therefore, Blockchain technology can ensure the data consistency and immutability.

2.3 Computing Technologies

Computing technologies such as cloud, edge, and fog computing enhance Generative AI by providing scalability, storage, real-time inference, bandwidth optimization, cost efficiency, privacy, and security. They enable efficient training, deployment, and utilization of Generative AI models, fostering innovation and advancing the field. Cloud computing provides access to vast computing resources, allowing for the scalability of Generative AI models. Training and generation tasks can be distributed across multiple servers or instances, enabling faster processing and accommodating larger models and datasets. Cloud computing provides ample storage capabilities for storing large datasets, model checkpoints, and generated content. This allows researchers and developers to access and manage their data efficiently. Contrary to the cloud computing, edge computing brings computational power closer to the data source or end-user device, enabling real-time inference for Generative AI applications. By processing data locally on edge devices, such as smartphones or IoT devices, latency is reduced, and real-time generation of content becomes feasible. This is especially useful for interactive applications like augmented reality, virtual reality, or real-time image and speech generation. Fog computing, which extends cloud computing capabilities to the network edge, allows for local processing and optimization of Generative AI workflows. By performing some preprocessing or feature extraction on edge devices or fog nodes, bandwidth consumption can be reduced, making efficient use of network resources. Cloud computing provides a cost-effective solution for Generative AI. Instead of investing in expensive on-premises infrastructure, organizations can leverage cloud services to access computing power on a pay-as-you-go basis, reducing upfront costs and allowing for flexibility in resource allocation. Edge and fog computing address privacy and security concerns in Generative AI. By processing data locally on edge devices or on edge servers within a local network, sensitive data can be kept within the premises and not transmitted to the cloud, enhancing data privacy and reducing security risks associated with data transfer.

2.4 Deep Learning

Deep Learning enhances Generative AI by enabling complex pattern recognition, representation learning, transfer learning, capturing long-term dependencies, and providing generative model architectures. These capabilities empower Generative AI models to generate more realistic, diverse, and contextually relevant content across various domains such as image generation, text generation, or audio synthesis.

Deep Learning models, such as recurrent neural networks (RNNs) or transformers, are effective in capturing long-term dependencies in sequential data. Transfer learning enables the models to transfer the learned features and representations from one domain or task to another, even with limited training data. Deep Learning models trained on large-scale datasets can leverage their learned knowledge to enhance generative capabilities. Transfer learning accelerates the training process and improves the quality of generated content by leveraging pre-trained models on related tasks or datasets.

Generative adversarial networks (GANs) and variational autoencoders (VAEs) are the generative model architectures which enable the generation of new content by learning the underlying distribution of the training data and sampling from it. The types of Deep Learning techniques, such as back-propagation and stochastic gradient descent, are used to train the generative models. This allows to capture complex patterns. Thereby, diverse and high-quality content can be generated.

Representation learning (unsupervised or semi-supervised learning) enables Deep Learning models to have the ability to learn meaningful representations of data. It allows the models to capture the underlying structure and features of the data, facilitating the generation of coherent and contextually relevant content which is crucial for Generative AI. Furthermore, Deep Learning models can learn hierarchical representations that results in the capability of generating content at different levels of abstraction.

Complex patterns and structures in data can be inferred by deep neural network. It allows the learning models to capture intricate relationships and generate more realistic and detailed content. Deep Learning models excel at understanding high-dimensional data (images, audio, and text). This enables them to generate diverse and visually appealing content.

2.5 Natural Language Processing

NLP techniques enable Generative AI models to understand and interpret human language. They extract meaning, identify entities, understand syntactic and semantic structures, and capture contextual information. This understanding is crucial for generating coherent and contextually relevant content.

NLP techniques play a vital role in generating natural language text. Generative AI models use algorithms such as language modeling, text generation, and sequence-to-sequence modeling to produce human-like sentences, paragraphs, or documents. NLP enables Generative AI models to understand and generate content in context. Models utilize contextual information from previous text to generate coherent and contextually appropriate responses. Techniques like contextual word embeddings and attention mechanisms aid in capturing and utilizing context effectively. NLP enables cross-lingual understanding, language transfer, and generation in diverse linguistic contexts.

2.6 Knowledge Graph

Knowledge Graphs (KG) capture rich semantic information and context about entities, their attributes, and relationships. Generative AI models can leverage this structured knowledge to enhance their understanding of the context in which they generate content. By incorporating information from the Knowledge Graph, the models can generate more accurate and contextually relevant content. Knowledge Graphs provide a comprehensive repository of structured information. Generative AI models can utilize this structured knowledge to improve their entity recognition and fact extraction capabilities. Generative AI models can utilize the information stored in the Knowledge Graph to generate content that aligns with existing knowledge. Generative AI models can utilize this information to expand and enrich the context of the generated content. By incorporating additional relevant information from the Knowledge Graph, the models can generate content that is more comprehensive and informative. Generative AI models trained with domain-specific Knowledge Graphs can generate domain specific content to generate content with domain-specific terminology, concepts, and relationships, making them more suitable for specialized applications.

2.7 Computer Vision

Computer Vision may enhance Generative AI in image understanding and generation. Image Processing techniques enable Generative AI models to analyze and understand visual data. By applying various image processing algorithms, such as image segmentation, object detection, and feature extraction, the models can extract meaningful information from images. This understanding of visual content helps in generating more contextually relevant and accurate image-based content. Image Generation utilize image synthesis methods combined with image processing algorithms to generate new images. These techniques help in creating visually appealing and visually coherent images that resemble real-world objects, scenes, or artwork.

Computer Vision techniques enable Image Data Augmentation by applying transformations or modifications to existing images. Techniques like image rotation, scaling, cropping, or adding noise can be used to generate new variations of training images. Image Data Augmentation enhances the diversity and robustness of the training data, improving the performance and generalization capabilities of the Generative AI models. These capabilities empower Generative AI models to generate visually appealing, realistic, and contextually relevant image-based content.

2.8 Immersive Technologies

Generative AI can generate realistic 3D models and visualizations of architectural designs. Combined with augmented reality (AR) technology, generative models can create immersive virtual experiences that allow stakeholders to visualize and interact with architectural designs prior to construction. This supports communication, collaboration, and decision-making processes to improve design understanding and reduce errors.

Delgado et al. [30] examine the current use of augmented reality (AR) and virtual reality (VR) in architecture, engineering, and construction. This identifies his six major use cases for AR and VR, including stakeholder engagement, design support, design review, construction support, operational and management support, and training. In this study, three main research categories are identified for future investigation: (i) technical devices focused on developing robust devices suitable for real-world construction site conditions, (ii) workflows and data management aimed at effectively managing the data and processes required for AR and VR technologies, and (iii) new features. AR is seen as a viable option to reduce data overload by allowing facility managers to visualize real-world environments using interactive data.

3 Generative AI Applications in the AEC-FM Industry

The AEC-FM industry with the requirements of multi-alternative outputs in a complex project environment with conflicting and large number of parameters may benefit from the Generative AI in design, construction and facility management activities. Generative AI can be used for generating multi design alternatives in architectural and engineering purposes, scheduling and budgeting alternatives for optimizing the construction process, aiding decision making in facility management.

The main purpose of using Generative AI is to enable machines to generate new and meaningful content similar to human-generated output. In the field of Generative AI, the focus is not only on mimicking or duplicating existing data, but also on generating new results that demonstrate creativity and imagination. Generative AI enables machines to learn from vast amounts of training data and use that knowledge

to create new, unique, and often surprising things. As data collection capabilities increase in the AEC-FM industry, the impact of big data and, in parallel, processing power technology in this area are directly understood.

In architectural design process, Image Processing techniques, such as style transfer algorithms, enable Generative AI models to transfer artistic styles from one image to another. This allows the models to generate designs for architects which can step further with designing “as a specific Pioneer architect”. This may lead to a retrospective of architectural design with the style of passed away pioneer architects. This leads to mimic the style of a particular architect or architectural style. Style transfer enhances the creativity and diversity of the generated images, enabling the models to produce visually unique and appealing content. Image Processing techniques can be used to restore or enhance the quality of images. Generative AI models can incorporate image restoration algorithms, such as denoising, deblurring, or inpainting, to improve the visual quality of images. This enhances the fidelity and clarity of the generated images, making them more visually appealing and informative.

Generative AI helps optimization of the energy efficiency and sustainability of buildings. By analyzing energy consumption patterns and environmental data, generative models can suggest design changes, such as optimizing building orientation, incorporating renewable energy sources, or suggesting energy-efficient systems and materials. This assists architects and engineers create greener and more sustainable buildings. GENE_ARCH [31] is a generative design system that follows an evolutionary approach to the design of building forms with the aim of creating energy efficient and sustainable building solutions. The system uses goal-oriented design, allowing architects to set performance goals for a building and search a specific design space for building solutions that meet those needs. GENE_ARCH uses the Pareto genetic algorithm as the search engine and DOE2.1E, an advanced building energy simulation program, as the scoring engine. The system runs a full DOE2.1E simulation for each of the thousands of alternative building solutions it generates, using real-world climate data for building locations, and runs hourly simulations throughout the year. Using DOE2.1E increases confidence in the results obtained from GENE_ARCH.

In architectural and engineering design process, Generative AI brings a level of automation to the design and construction process that enables AEC professionals to create optimal designs rapidly and dramatically reduces construction errors. The industry uses AI-powered generative design to identify and collaborate the design of the various teams to ensure zero clashes which in turn mitigate the risk of rework. Generative design of engineering outputs may enhance a large decrease in construction waste to serve lean construction principles. Consequently, construction costs and time decrease without any scope creep. In summary, Generative Design, has potential outputs in conceptual design exploration, form and shape optimization, parametric design, material selection and optimization, site analysis and contextual design, design compliance and code checking (Fig. 1).

Generative Design (GD) is an iterative design process based on a set of rules or automatically explores, iterates, and optimizes design options using algorithms and parametric modeling.

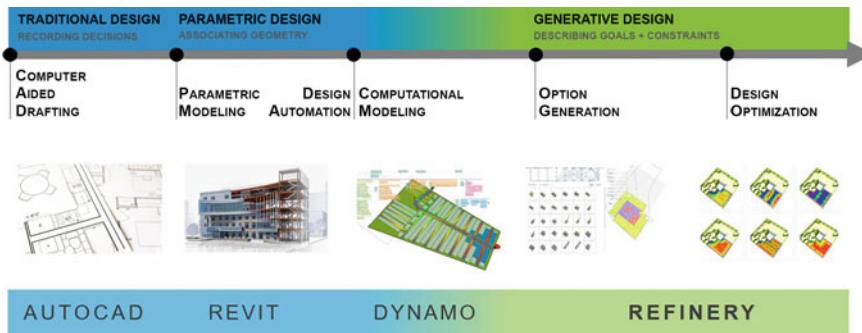


Fig. 1 Adopted from: <https://medium.com/autodesk-university/from-design-automation-to-generative-design-in-aec-8b3c0b1078db>

Designs are analyzed and evaluated through multiple iterations against specific goals such as structural integrity, energy efficiency, cost efficiency, or user comfort. Generative AI can learn from data and performance feedback, suggest design changes, and automatically optimize designs based on certain criteria via machine learning algorithms. Shea et al. [32] describe the combination of GD techniques and associative modeling systems to facilitate the development of meaningful input models for structural GD. The integration of GD methodologies and associative modeling systems aims to inspire creative solutions and enable designers to develop robust models of design constraints, language, and performance. Joe and Pelosi [33] explores the potential of collaborative, human-generative design tools in optimizing building performance and enabling large-scale prefabricated housing while promoting design variance. Chan et al. [34] argued that the gap between automation and optimization should be addressed as a complex combinatorial optimization problem for layout design and should focus on computational techniques that could potentially support automation and optimization. They state that using genetic algorithms as part of an evolutionary design approach offers a promising way to automate and optimize the complex design process.

In construction process, Generative AI can contribute to construction planning and optimization by simulating and generating construction processes and schedules. By considering factors such as site constraints, resource availability, and project goals, generative models can propose optimized construction plans, material allocation strategies, and logistics plans. This can shorten project schedules, minimize costs, and improve construction efficiency. There are not many Generative AI implementations in the AEC-FM industry. However, the possible positive impacts may occur in planning, scheduling, resource allocation, resource management, cost estimation, budgeting, safety planning, risk assessment, prefabrication, modular construction, quality control, energy efficiency, and sustainability. Further there may be on site organization optimization, communication, and collaboration with Generative AI solutions.

In facility management process, AI may reduce costs by ensuring more efficient facility management. It also improves quality and reduces downtimes as it recommends the best way to react to anomalies. Through improved deep learning and machine learning, Generative AI could be used to draw upon existing data sets and product information for a quicker, more efficient facility management process. The use cases in facility management may be space allocation, space optimization, maintenance planning, predictive maintenance, energy management, fault detection, fault diagnosis, security, and safety optimization by supporting data-driven decision making. In facility management, Generative AI helps optimization of space planning and layout. By considering factors such as occupancy patterns, workflow efficiency, safety regulations, and user preferences, Generative AI algorithms can suggest optimal layouts for offices, hospitals, retail spaces, and other built environments. Zhao et al. [35] address challenges in hospital building design and proposes smart design techniques for the design of a general hospital surgical department. Traditional layout design methods are labor intensive, time consuming and error prone. In response, Zhao et al.'s research explores the application of artificial intelligence (AI) in the design of hospital layouts, especially using systematic health layout planning and generative adversarial network (GAN) approaches.

Generative AI can optimize the performance of building systems such as HVAC (heating, ventilation, and air conditioning), lighting, and energy management systems. By analyzing data on energy usage, environmental conditions, and user behavior, Generative AI can suggest system configurations and control strategies that maximize energy efficiency, occupant comfort, and operational cost savings. By analyzing historical data, generative models can predict potential equipment failures, optimize maintenance schedules, and suggest maintenance interventions to minimize downtime and costs. This can improve the efficiency and longevity of building systems and infrastructure. By analyzing energy consumption patterns and environmental data, generative models can suggest design changes, such as optimizing building orientation, incorporating renewable energy sources, or suggesting energy-efficient systems and materials.

4 BIM as a Generative AI Facilitator for the AEC-FM Industry Applications

Building Information Modeling (BIM) is a digital representation of a building or infrastructure project, including geometry, spatial relationships, and other relevant data. Generative AI can contribute to BIM workflows by automatically generating or enhancing BIM data. For example, generative models can generate 3D models from 2D drawings, extract information from point cloud data, and automatically annotate and classify elements in BIM models.

Generative AI facilitates parametric design, where designs are created using defined parameters and rules. By specifying variables and relationships such as

building dimensions, material properties, and structural requirements, Generative AI algorithms can generate designs that comply with these parameters. This allows architects to explore design variations easily and efficiently and also work with BIM in an efficient way. Abrishami et al. [36] introduces the G-BIM (Generative Building Information Modeling) workspace as a new technology. This has the potential to leverage conceptual design innovations in AEC projects. It builds on the results of initial research and provides a framework for using generative BIM workspaces during the design phase. This framework emphasizes the connections and dependencies between generation/parametric tools and BIM applications, accelerating information transfer using generation tools based primarily on neutral BIM standards. The results show that applying GD can greatly improve the design experience by assisting the designer in the iterative generation of design alternatives and parameterization processes. This framework specifically integrates BIM and GD to improve the design process during the concept design stage. This forms the basis for a working prototype that actively incorporates the GD methodology into a single dynamic BIM environment. Hamidavi et al. [37] emphasize the importance of integrated automated structural design, especially for high-rise buildings, in the early stages of a project. They propose to use automated prototyping within a BIM-based platform to facilitate structural design, analysis and optimization. Hamidavi et al.'s study suggests using Robot Autodesk to generate various structural model options. Their purpose is to expand the scope of automation within BIM platforms, reduce repetitive processes in areas such as conceptual structural design, and improve collaboration between architects and civil engineers. Automating these processes is expected to save time and money in the early stages of a project. As stated by Ma et al. [38], the integration of generative design (GD) and building information modeling (BIM) has emerged as a promising technology consolidation. This technology is called as GD-BIM. Ma et al. [38] critically review the existing approach to GD development in a BIM framework. The research proposes new perspectives, including objective-oriented, GD component-based, and skill-driven GD-BIM development, along with reference guides. They emphasize the need for methodological guidance to assist designers, especially those with little knowledge of GD or programming, in developing GD within the BIM context.

Wei et al. [39] focused on the use of generative design and Building Information Modeling (BIM) to automate the modular construction of residential buildings in urban areas. The purpose of Wei et al.'s research is to develop a systematic approach to parameterize design parameters and integrate design rules into modular building layout design within a BIM environment. The proposed approach uses generative design to generate recommendations and analytical calculations to optimize the construction process by considering building regulations and user input. The integration of BIM and generative design enables practitioners to rapidly create design layouts and assess constructability. This study combines construction planning and optimization focused generative design with BIM technology.

Gan [40] developed a systems approach to optimizing carbon footprint and construction costs in off-site construction using building information modeling

(BIM) and generative design. The study proposes a mathematical formula representing the topological relationship between various precast and cast-in-place elements, enabling the 3D geometric modeling and variations within precast buildings. New generative algorithms are being developed to manipulate building geometry and create parametric BIM models based on user-defined constraints. An automated tool is created that extracts the model's geometry and matches it against a custom BIM object library to analyze sustainability factors such as carbon emissions and cost factors. This approach enables a comprehensive investigation of alternative designs in precast construction. The study includes case studies demonstrating the impact of different design options on CO₂ emissions and construction costs, showing significant reductions compared to reference buildings. The proposed method enables automated design exploration at an early stage, facilitating informed energy efficiency decision-making towards optimal solutions via BIM through Generative AI.

5 Gaps and Trends for Generative AI Research in the AEC-FM Industry

The bibliometric search was performed via Boolean Syntax in Scopus using {generative} AND {artificial intelligence} OR {AI} OR {NLP} OR {natural language processing} OR {computer vision} OR {image processing} OR {computing} OR {Augmented reality} OR {virtual reality} OR {deep learning} OR {distributed ledger technology} ..." keywords and additional limitations. The search results with predefined keywords in Scopus were found to be 557 papers. Finally, the number of papers was further reduced to 125 articles with additional limitations to focus on the directly related researches. The final number of papers considered for the scientometric analysis and mapping was 115 articles.

The number of articles published between 2002 and 2023 (till June 2023) are displayed yearly in Fig. 2. The number of papers related to Generative AI research in the AEC-FM industry has gone up considerably in this period. Especially in 2018, the number of papers displays a critical upturn. The average number of papers in the last five years (2019–2023) is around 82 (the number of papers in 2023 could not be completely monitored by the time the paper was prepared). Trend of the research in the Generative AI research in the AEC-FM industry shows an exponential growth, which highlights a major potential impact on the AEC-FM community.

Figure 3 presents the number of articles in Generative AI research in the AEC-FM industry from 2002 to 2023 by country. According to the article numbers, one may affirm that the most influencing country in the research area is the United States followed by the United Kingdom and China. Being the pioneer countries in Generative AI research in the AEC-FM industry may be the result of being pioneer in advanced technology implementations in the industry, supporting comprehensive research projects, and being innovation leaders of the related technologies.

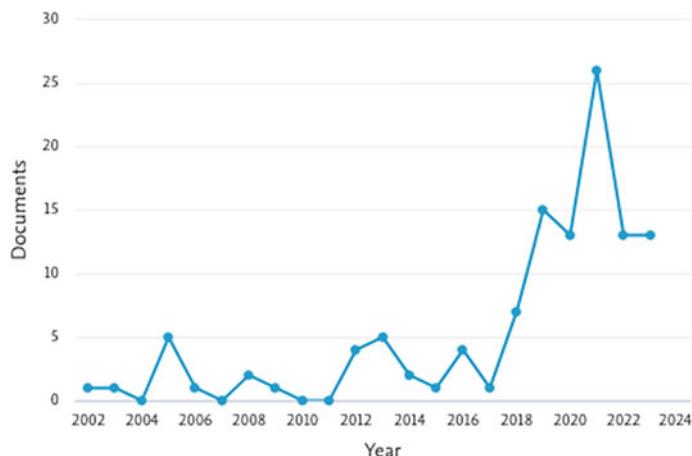


Fig. 2 The yearly number of articles in Generative AI research in the AEC-FM industry from 2002 to 2023

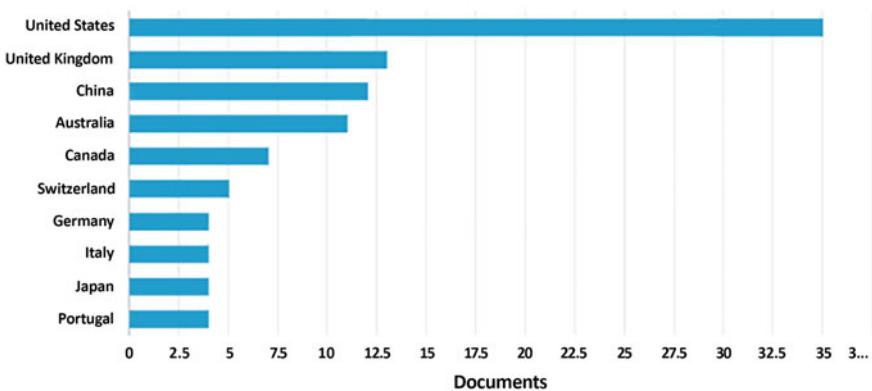


Fig. 3 The number of articles in Generative AI research in the AEC-FM industry from 2002 to 2023 by country

5.1 Keyword Co-occurrence Analysis for Generative AI Use in the AEC-FM Industry

VOSviewer is a scientometric analysis tool that is used for analyzing bibliometric network data. He et al. [41] consider the keywords to be the most essential elements in the detection of key aspects of a research area. A network of keywords emphasizes the relationship among each other and the knowledge organization of the major research areas [42]. The co-occurrence of keywords indicates the number of articles in which they both occur. Bibliometric data was acquired from Scopus and used to construct a keyword co-occurrence analysis [43]. The analysis was performed for

the period between 2002 and 2023 (till June 2023). The use of index keywords as a searching strategy eliminates the risk of missing any keyword. Therefore, index and author keywords were used together for the analysis. The focus of the paper can be understood by the number of a keyword in a document. Hence, full counting is used for the analysis. The map in Fig. 4 is based on the collected bibliometric data. 902 words were analyzed. The minimum number of co-occurrence of a keyword was set to 3. The number of keywords met that the threshold was 58.

The result of the co-occurrence analysis of keywords is mapped and illustrated with a network diagram in Fig. 4. The color of a keyword determines the cluster to which the keyword belongs. The keywords were clustered by colors in six major categories (Fig. 4). The clusters were named after the field of study that the keyword is belonged to (Table 2) such as '*Lifecycle management with generative AI*', '*Predictive modeling, operations, and maintenance with generative AI*', '*Machine learning implementation in design*', '*Generative AI in design optimization*', '*Constraint-based generative design*', and '*Design techniques in generative AI*'.

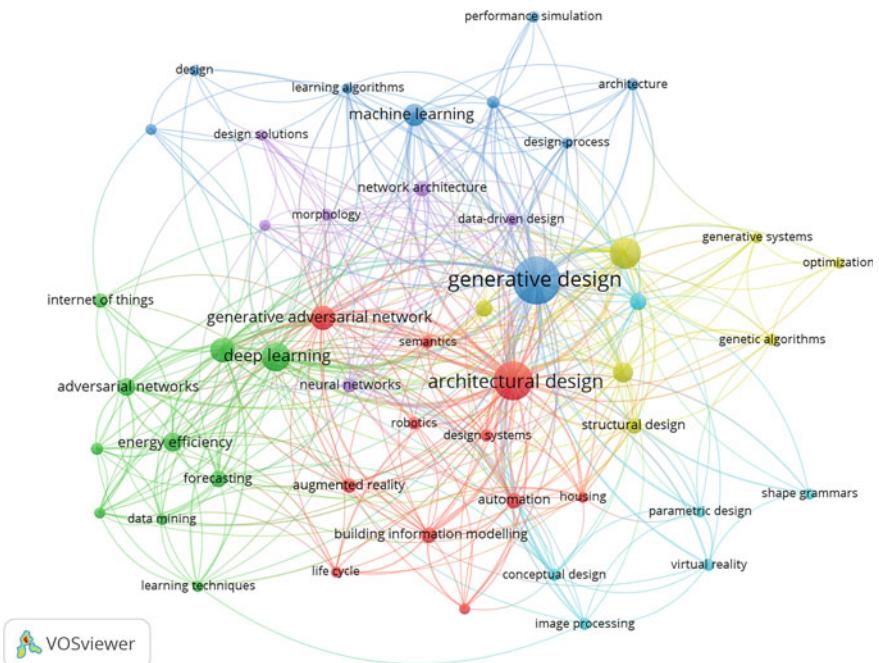


Fig. 4 The network map visualization of the co-occurrence of author and index keywords focused on Generative AI research in the AEC-FM industry between 2002 and 2023

Table 2 Generative AI research areas in the AEC-FM industry

Keyword group	References
<i>Cluster 1—Lifecycle management with generative AI</i>	
Architectural design	[44, 45]
Augmented reality	[46–48]
Automation	[46, 49–51]
Building information modeling	[36, 46, 51–53]
Design systems	[36, 51, 54–59]
Generative adversarial networks	[35, 47, 50–52, 60–63]
Housing	[48, 64, 65]
Information management	[36, 46, 66]
Life cycle	[51, 67, 68]
Robotics	[69],
Semantics	[47, 59, 60, 70]
<i>Cluster 2—Predictive modeling, operations, and maintenance with generative AI</i>	
Data mining	
Deep learning	[35, 47, 50, 61, 62, 71–75]
Energy efficiency	[50, 61, 62, 65, 72]
Forecasting	[61, 72]
Intelligent buildings	[50, 63]
Internet of things	[76–78]
Learning systems	[71]
Learning techniques	[46, 50, 73, 75]
Predictive analytics	[48, 50, 61, 77, 79]
<i>Cluster 3—Machine learning implementation in design</i>	
Design	[52, 53, 80]
Design-process	[45, 51, 81]
Generative design	[36, 44, 53, 54, 73, 82]
Learning algorithms	[69, 73, 74]
Machine learning	[45, 52, 71, 80, 81]
Performance simulation	[45, 83]
Reinforcement learning	[73, 84, 85]
<i>Cluster 4—Generative AI in design optimization</i>	
Artificial intelligence	[54, 65, 81]
Computer aided design	[80]
Deep neural networks	[48, 49, 62, 71, 86–88]
Generative systems	[55, 57, 84, 86, 89]
Genetic algorithms	[36, 44, 64, 90–92]

(continued)

Table 2 (continued)

Keyword group	References
Optimization	[51, 55, 56, 65, 80, 93]
Structural design	[45, 48, 49, 93, 94]
<i>Cluster 5—Constraint-based generative design</i>	
Data-driven design	[50, 71, 95]
Design solutions	[51, 81]
Generative model	[48, 64, 72, 93, 96–98]
Morphology	[70, 75, 98, 99]
Network architecture	[49, 77]
Neural networks	[44, 48, 55, 65, 75, 80, 92, 98]
<i>Cluster 6—Design techniques in generative AI</i>	
Computational design	[56, 71, 81]
Conceptual design	[51, 65, 71]
Image processing	[75]
Parametric design	[58, 82]
Shape grammars	[44, 100]
Virtual reality	[44, 78, 101, 102]

5.1.1 Lifecycle Management with Generative AI

Getuli et al. [103] aim to improve workplace planning in construction projects using immersive and building information modeling (BIM) technologies. This approach is to take knowledge from experienced workers and integrate it with the expertise of field managers. This research uses real case studies to demonstrate the benefits of this approach, demonstrating improved exchange of planning and safety-related information between project partners. The complexity of construction projects and the need for effective information sharing pose challenges, highlighting the importance of collecting and integrating information from multiple stakeholders. Accurately defining the work area is very important for efficient and safe planning of construction measures. The research hypothesis suggests that advanced technologies such as BIM and VR can be shared to improve workplace planning, safety information sharing, and project communication between partners. The research aims to create a multi-dimensional project model that integrates workplace planning, safety-related information, and planning of construction activities through the use of BIM and VR technology. The study recognizes the important role played by the experience and knowledge of field managers, as well as input from subcontractors and workers involved in the activity. The hypothesis is that the use of advanced information technologies such as BIM and VR will improve the accuracy and reliability of workplace definitions, resulting in more realistic project planning, safety and planning information to all involved partners. This integrated approach has the potential to improve

collaboration in the construction industry, facilitate information sharing, and improve overall project outcomes.

5.1.2 Predictive Modeling, Operations, and Maintenance with Generative AI

In the proposed intelligent functional layout generation method for emergency departments, Zhao et al. [35] used GAN to generate new emergency department layouts based on a dataset of existing emergency department floor plans. The GAN model is trained to generate new layouts that are similar to the existing ones, but also have some degree of variation and creativity. The GAN model is trained to generate each level of the layout, and the results are compared from the perspectives of architecture and algorithm to increase the applicability of the entire process. The Generative AI's contribution to the proposed intelligent functional layout generation method for emergency departments is to provide a way to generate new layouts that are similar to existing ones, but also have some degree of variation and creativity. This can help architects and designers to explore new design possibilities and come up with more innovative and efficient emergency department layouts.

Generative AI, as utilized Godahewa et al. [72], plays a significant role in optimizing energy consumption. Generative models have the ability to generate new data samples that resemble the training data distribution. In the context of energy consumption optimization, Generative AI models can generate synthetic time series data for unoccupied periods of a room, allowing for more accurate predictions of future temperatures. By incorporating Generative AI into the deep learning framework, the researchers are able to address the cold-start problem, where there is a lack of historical data for certain periods. This is particularly useful in scenarios where there is limited or no data available for unoccupied periods. The generative model can generate realistic temperature sequences for these periods, enabling more accurate predictions and better optimization of setpoints. Thereby, the Generative AI component of the framework helps to improve the accuracy of temperature predictions during unoccupied periods, leading to more effective optimization of energy consumption in air conditioning systems.

In the context of predictive maintenance, the GAN was used to generate synthetic data that represents the normal and abnormal states of the machine [79]. This synthetic data can be used to train the predictive maintenance model and improve its accuracy in detecting faults and predicting maintenance needs. The GAN can also help to avoid the mode collapse of the generative model and realize the self-detection of abnormal data. As a result, the Generative AI implementation enabled predictive maintenance based on deep adversarial learning to improve the accuracy and efficiency of prognostics and maintenance tasks in the intelligent manufacturing system.

5.1.3 Machine Learning Implementation in Design

Wang et al. [104] discuss the relationship between urban planning and machine learning. It sheds light on various machine learning concepts, including adversarial learning, generative neural networks, encoder/decoder deep networks, conversational AI, and geospatial and temporal machine learning. The focus is on how AI can contribute to modern urban planning, especially in automated land-use configuration. This includes generating land use and building configurations based on geospatial data, human movements, social media, environmental factors, and economic activity. The study concludes with an overview of the impact of AI on urban planning, proposing areas of research that overlap in both areas.

Generative AI can play a role in optimizing traffic flow in urban planning, especially in dealing with congestion and delays. By analyzing real-time traffic data and predicting traffic patterns, Generative AI can help improve traffic flow in metropolitan areas. Adaptive traffic light control and dynamic lane assignment are two key strategies in traffic management that can benefit from Generative AI. Adaptive traffic light control helps reduce congestion and improve traffic flow by adjusting the timing of traffic lights in real time based on sensors and real-time traffic data. Dynamic lane assignment temporarily changes lane direction during rush hour to optimize traffic flow. Adaptive cruise control (ACC) systems are becoming more common in new smart vehicles and will play a key role in the future automated driving. ACC enables the vehicle to automatically maintain a distance to the vehicle ahead. Studies show that a shorter separation distance improves traffic throughput, and effective use of ACC can improve traffic flow by adjusting the desired separation distance according to changing traffic conditions. However, Elmorshedy et al. [105] show that even with short transit times, traffic congestion can still occur if the bottleneck density exceeds a critical value. To address this issue, researchers propose adaptive gain learning (RL) proximity controllers that use ACC to optimize traffic flows and minimize delays. Using deep reinforcement learning (DRL), the controller dynamically assigns optimal distance values to different sections of the highway, considering highway and ramp traffic conditions. Simulation results show that this dynamic distance control strategy significantly improves traffic flow and reduces system delay by up to 57% compared to the fixed distance strategy.

5.1.4 Generative AI in Design Optimization

Merabet et al. [106] provide a comprehensive overview of AI techniques used in building control systems to optimize energy use while maintaining thermal comfort. This study examines various AI techniques used in this field and evaluates their effectiveness in improving energy efficiency and user comfort. They expose the complexities of balancing energy use and comfort requirements in buildings and provides insights to help researchers and professionals address this challenge. An analysis of 20 AI tools developed for energy consumption and comfort control shows capabilities such as pattern recognition, optimization and predictive control. The

study recognizes that while the use of AI in building management is promising, further improvements are needed, mainly due to the lack of high-quality real-world data in the energy sector. The results show that combining AI technology with a personalized comfort model resulted in significant energy savings and increased comfort.

5.1.5 Constraint-Based Generative Design

Nagy et al. [107] illustrate the application of generative design at a city scale using the example of a real-life neighborhood development project in Alkmaar, the Netherlands. The urban planning problem is known for its complexity and the involvement of multiple stakeholders with different and potentially conflicting needs. In this research, they demonstrate the potential of generative design to address this complexity by optimizing two main goals: The profitability of the project for the developer, the power generation potential of the solar panels on the roof of the building. This study highlights the need for further research to explore the application of generative design in solving urban-scale design problems.

5.1.6 Design Techniques in Generative AI

Sydora and Stroulia [53] address the need for automating the evaluation and generation of interior designs using BIM. By introducing a domain-specific language and corresponding algorithms, the study enables the representation and evaluation of design rules for interior spaces. The proposed rule language is flexible and expressive, allowing for complex rules based on geometrical properties and relations. The research demonstrates the application of the language in two case studies, showcasing its usefulness in generating multiple code-compliant designs. Additionally, a software toolkit is provided, offering RESTful APIs for easy integration with external applications. The research findings contribute to the automation and optimization of the design process in the construction industry, specifically in relation to interior space planning and compliance checking using BIM.

6 Future Prospects for Generative AI in the AEC-FM Activities

In terms of the insights on how Generative AI can enhance various aspects of the AEC-FM industry are stated below for a prospective perspective with clustering major topics of *lifecycle management with generative AI, predictive, modelling, operations, and maintenance with generative AI, machine learning implementation*

in design, generative AI in design optimization, constrained-based generative design, and design techniques in generative design.

Generative AI has the potential to revolutionize design processes by offering intelligent design support. Generative models can generate design proposals that optimize designs against multiple criteria by analyzing project requirements such as design constraints resourced by the style, site, codes, and client's requirements; and historical data that is transferred as lessons learned from previous projects. This empowers construction project design team (architects and engineers) to make well-informed decisions, resulting in more efficient and innovative designs. City-level planning can benefit from Generative AI through generative models which can generate alternative layouts, transportation systems, and zoning plans by leveraging big data. These models optimize factors such as transportation efficiency, sustainability, and quality of life, aiding in the design and optimization of cities and urban spaces. Along with Digital Twins integration Generative AI can boost the smart city and intelligent building implementation and performance. Generative AI can enhance the design and simulation experience with immersive technologies. Generative models enable construction project stakeholders to explore real-world experience, visualize design concepts, and experience spatial qualities through immersive technology integrated BIM models before construction. This leads to a higher level of design understanding, useful feedback, and improved data-driven decision-making processes. Generative AI can streamline construction planning. By generating optimized construction schedules, schedule simulations, and material allocation plans, generative models consider factors such as site constraints, resource availability, and project goals. In turn project timelines are reduced, costs are minimized, and construction process' efficiency increases. Generative AI is used for analyzing data from drones, cameras, and sensors. Generative models can automatically detect anomalies, identify potential safety risks, and monitor construction progress. Thereby, another contribution of Generative AI in AEC-FM activities may be autonomous monitoring and controlling of construction sites and built environment. This improves safety, quality control, and project management, effectively addressing industry challenges related to inefficiency and low productivity. As in other contributions, Generative AI can also play a crucial role in the lifecycle management and sustainability of buildings and infrastructures. Generative models can optimize building operations, predict maintenance needs, and suggest sustainable retrofitting strategies by analyzing historical and real-time data on energy usage, maintenance records, and environmental impact. This enables predictive and proactive management, reduces energy consumption, and promotes sustainable practices in the AEC-FM industry. Generative AI, can unlock immense potential for efficiency, innovation, and sustainability across various aspects of AEC-FM activities. The most majority of the Generative AI researches in the AEC-FM industry uses Generative Adversarial Networks. This is most probably caused by the model's relatively low need for big data. GANs enhance generation of realistic data, can learn to generate data without explicit labels and realistic data samples, can generate entirely new and novel samples, can augment training data, can transfer knowledge from one domain to another. However, GANs have shortfalls in stable training, generator durability, quantitative evaluating the performance,

optimal computation time, and hyperparameter sensitivity. Besides, there are other Generative AI models can be used in AEC-FM activities such as Deep Convolutional Generative Adversarial Network (DCGAN), Variational Autoencoder (VAE), Conditional Variational Autoencoder (CVAE), StyleGAN, CycleGAN, DeepArt, Pix2Pix, Autoencoder Variational Bayes (VAE), Generative Pre-trained Transformer (GPT) and others. Each model can successfully ease and accelerate AEC-FM activities. Therefore, future researches may benefit from above-mentioned models.

As the technology continues to develop, more innovative and groundbreaking applications will improve the way the design, construction, and management of buildings and facilities. Generative AI is expected to enhance a better state of the art solutions in terms of *automation*, *sustainability*, and *resilience* in the AEC-FM industry related research and activities.

References

1. Zhang, Q., Song, Y., Jiao, P., & Hu, Y. (2022). A Hybrid and hierarchical approach for spatial exploration in dynamic environments. *Electronics*, 11(4), 574.
2. Raji, R. S. (1994). Smart networks for control. *IEEE Spectrum*, 31(6), 49–55.
3. Allam, Z., & Dhunny, Z. A. (2019). On big data, artificial intelligence and smart cities. *Cities*, 89, 80–91.
4. Cheng, J. C., Chen, W., Chen, K., & Wang, Q. (2020). Data-driven predictive maintenance planning framework for MEP components based on BIM and IoT using machine learning algorithms. *Automation in Construction*, 112, 103087.
5. Costin, A., Wehle, A., & Adibfar, A. (2019). Leading indicators—A conceptual IoT-based framework to produce active leading indicators for construction safety. *Safety*, 5(4), 86.
6. Kong, L., & Ma, B. (2020). Intelligent manufacturing model of construction industry based on Internet of Things technology. *The International Journal of Advanced Manufacturing Technology*, 107, 1025–1037.
7. Tang, S., Shelden, D. R., Eastman, C. M., Pishdad-Bozorgi, P., & Gao, X. (2019). A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends. *Automation in Construction*, 101, 127–139.
8. Carmona, A. M., Chaparro, A. I., Velásquez, R., Botero-Valecia, J., Castano-Londono, L., Marquez-Viloria, D., & Mesa, A. M. (2019). *Instrumentation and data collection methodology to enhance productivity in construction sites using embedded systems and IoT technologies*. Advances in Informatics and Computing in Civil and Construction Engineering: Proceedings of the 35th CIB W78 2018 Conference: IT in Design, Construction, and Management, pp. 637–644. Springer International Publishing.
9. Joel, M. R., Ebenezer, V., Karthik, N., & Rajkumar, K. (2019). Advance dynamic network system of internet of things. *International Journal of Recent Technology and Engineering*, 8(3), 6209–6212.
10. Lee, I., & Lee, K. (2015). The internet of things (IoT): applications, investments, and challenges for enterprises. *Business Horizons*, 58(4), 431–440.
11. Niu, S., Pan, W., & Zhao, Y. (2016). A virtual reality integrated design approach to improving occupancy information integrity for closing the building energy performance gap. *Sustainable Cities and Society*, 27, 275–286.
12. Gamil, Y., A. Abdullah, M., Abd Rahman, I., & Asad, M. M. (2020). Internet of things in construction industry revolution 4.0: Recent trends and challenges in the Malaysian context. *Journal of Engineering, Design and Technology*, 18(5), 1091–1102.

13. He, W., Li, S., & Xu, L. D. (2014). Internet of Things in industries: A survey. *IEEE Transactions on Industrial Informatics*, 10(4), 2233–2243. <https://doi.org/10.1109/TII.2014.2300753>
14. Opentechdiary (2015). *Internet of things world Europe*. Retrieved from: <https://opentechdiary.wordpress.com/2015/07/16/a-walk-through-internet-of-things-iot-basics-part-2/>.
15. Sung, W. T., Hsiao, S. J., & Shih, J. A. (2019). Construction of indoor thermal comfort environmental monitoring system based on the IoT architecture. *Journal of Sensors*.
16. Zhao, L., Liu, Z., & Mbachu, J. (2019). Development of intelligent prefabs using IoT technology to improve the performance of prefabricated construction projects. *Sensors*, 19(19), 4131.
17. Nagashree, R. N., Rao, V., & Aswini, N. (2014). Near field communication. *International Journal of Wireless and Microwave Technologies (IJWMT)*, 4, 20–30. <https://doi.org/10.5815/ijwmt.2014.02.03>
18. Peng, C., & Huang, J. (2016). A home energy monitoring and control system based on ZigBee technology. *International Journal of Green Energy*, 13(15), 16151623. <https://doi.org/10.1080/15435075.2016.1188102>
19. Salleh, A., Aziz, A., Abidin, M. Z., Misran, M. H., & Mohamad, N. R. (2013). Development of greenhouse monitoring using wireless sensor network through ZigBee technology. *International Journal of Engineering Science Invention, ISSN*, 2(7), 6–12.
20. Talaviya, G., Ramteke, R., & Shete, A. K. (2013). Wireless fingerprint-based college attendance system using Zigbee technology. *International Journal of Security, Privacy and Trust Management (IJSPTM)*, 5(4), 11–17.
21. Wang, W., He, G., & Wan, J. (2011, September 16–18). *Research on Zigbee wireless communication technology*. Proceedings of the 2011 International Conference on Electrical and Control Engineering (ICECE), Yichang, China, 1245–1249. <https://doi.org/10.1109/ICECENG.2011.6057961>
22. Whitmore, A., Agarwal, A., & Da Xu, L. (2015). The Internet of things—A survey of topics and trends. *Information Systems Frontiers*, 17, 261–274. <https://doi.org/10.1007/s10796-014-9489-2>
23. Zhang, D., Yang, L. T., Chen, M., Zhao, S., Guo, M., & Zhang, Y. (2016). Real-time locating systems using active RFID for Internet of Things. *IEEE Systems Journals*, 10(3), 1226–1235. <https://doi.org/10.1109/JSYST.2014.2346625>
24. Zillner, T. (2018). *Zigbee exploited—the good, the bad, the ugly*. Available online: <https://www.blackhat.com/docs/us-15/materials/us-15-Zillner-ZigBeeExploited-The-Good-The-Bad-And-The-Ugly.pdf>. Accessed on 6 January 2018.
25. Häikiö, J., Kallio, J., Mäkelä, S. M., & Keränen, J. (2020). IoT-based safety monitoring from the perspective of construction site workers. *International Journal of Occupational and Environmental Safety*, 4(1), 1–14.
26. Atlam, H. F., & Wills, G. B. (2020). IoT security, privacy, safety and ethics. In M. Farsi, A. Daneshkhah, A. Hosseiniyan-Far, & H. Jahankhani (Eds.). *Digital twin technologies and smart cities. Internet of things (technology, communications and computing)*. Springer. https://doi.org/10.1007/978-3-030-187323_8
27. Noor, M. B. M., & Hassan, W. H. (2019). Current research on Internet of Things (IoT) security: A survey. *Computer Networks*, 148, 283–294. <https://doi.org/10.1016/j.comnet.2018.11.025>
28. Dave, B., Buda, A., Nurminen, A., & Främling, K. (2018). A framework for integrating BIM and IoT through open standards. *Automation in Construction*, 95, 35–45.
29. Ozturk, G. B. (2021). The evolution of building information model: Cognitive technologies integration for digital twin procreation. In *BIM-enabled cognitive computing for smart built environment* (pp. 69–94). CRC Press.
30. Delgado, J. M. D., Oyedele, L., Demian, P., & Beach, T. (2020). A research agenda for augmented and virtual reality in architecture, engineering and construction. *Advanced Engineering Informatics*, 45, 101122.
31. Caldas, L. (2008). Generation of energy-efficient architecture solutions applying GENE-ARCH: An evolution-based generative design system. *Advanced Engineering Informatics*, 22(1), 59–70.

32. Shea, K., Aish, R., & Gourtovaia, M. (2005). Towards integrated performance-driven generative design tools. *Automation in Construction*, 14(2), 253–264.
33. Joe, J., & Pelosi, A. (2021). PARAMTR V2. (2021)
34. Chan, K. Y., Kwong, C. K., Dillon, T. S., & Fung, K. Y. (2011). An intelligent fuzzy regression approach for affective product design that captures nonlinearity and fuzziness. *Journal of Engineering Design*, 22(8), 523–542.
35. Zhao, C. W., Yang, J., & Li, J. (2021). Generation of hospital emergency department layouts based on generative adversarial networks. *Journal of Building Engineering*, 43, 102539.
36. Abrishami, S., Goulding, J., & Rahimian, F. (2021). Generative BIM workspace for AEC conceptual design automation: Prototype development. *Engineering, Construction and Architectural Management*, 28(2), 482–509.
37. Hamidavi, T., Abrishami, S., Ponterosso, P., Begg, D., & Nanos, N. (2020). OSD: A framework for the early stage parametric optimisation of the structural design in BIM-based platform. *Construction Innovation*.
38. Ma, W., Wang, X., Wang, J., Xiang, X., & Sun, J. (2021). Generative design in building information modelling (BIM): Approaches and requirements. *Sensors*, 21(16), 5439.
39. Wei, Y., Choi, H., & Lei, Z. (2022). A generative design approach for modular construction in congested urban areas. *Smart and Sustainable Building Environment*, 11(4), 1163–1181.
40. Gan, V. J. (2022). BIM-based building geometric modeling and automatic generative design for sustainable offsite construction. *Journal of Construction Engineering and Management*, 148(10), 04022111.
41. He, Q., Wang, G., Luo, L., Shi, Q., Xie, J., & Meng, X. (2017). Mapping the managerial areas of Building Information Modeling (BIM) using scientometric analysis. *International Journal of Project Management*, 35(4), 670–685.
42. Van Eck, N. J., & Waltman, L. (2014). Visualizing bibliometric networks. In *Measuring scholarly impact: Methods and practice* (pp. 285–320).
43. Ozturk, G. B. (2020). Interoperability in building information modeling for AECO/FM industry. *Automation in Construction*, 113, 103122.
44. Singh, V., & Gu, N. (2012). Towards an integrated generative design framework. *Design Studies*, 33(2), 185–207.
45. Yazici, S. (2020, September). *A machine-learning model driven by geometry, material and structural performance data in architectural design process*. Proceedings of the 38th eCAADe Conference, Berlin, Germany, pp. 16–18.
46. Chen, J., Li, S., Lu, W., Liu, D., Hu, D., & Tang, M. (2021, November). *Markerless augmented reality for facility management: Automated spatial registration based on style transfer generative network*. Proceedings of the 38th International Symposium on Automation and Robotics in Construction (ISARC), pp. 467–474.
47. Nakhaee, A., & Paydar, A. (2023, March). DeepRadiation: An intelligent augmented reality platform for predicting urban energy performance just through 360 panoramic streetscape images utilizing various deep learning models. In *Building simulation* (Vol. 16, No. 3, pp. 499–510). Tsinghua University Press.
48. Newton, D. (2019). Deep generative learning for the generation and analysis of architectural plans with small datasets. In *Proceedings of 37th eCAADe and 23rd SIGraDi Conference* (Vol. 2, pp. 21–28).
49. de Miguel, J., Villafane, M. E., Piskorec, L., & Sancho-Caparrini, F. (2019). *Deep form finding using variational autoencoders for deep form finding of structural typologies*. 37th Conference on Education and Research in Computer Aided Architectural Design in Europe (eCAADe) & 23rd Conference of the Iberoamerican Society Digital Graphics (SIGraDi), pp. 71–80. eCAADe-European Association for Education and Research in CAAD in Europe.
50. Fan, C., Sun, Y., Zhao, Y., Song, M., & Wang, J. (2019). Deep learning-based feature engineering methods for improved building energy prediction. *Applied Energy*, 240, 35–45.
51. Ghannad, P., & Lee, Y. C. (2021). Developing an advanced automated modular housing design system using deep learning and building information modeling (BIM). In *Computing in civil engineering 2021* (pp. 587–595).

52. Celik, F., Faltin, B., & König, M. (2021). Improving 2D construction plans with cycle-consistent generative adversarial networks. In *Computing in civil engineering* (pp. 50–57).
53. Sydora, C., & Stroulia, E. (2020). Rule-based compliance checking and generative design for building interiors using BIM. *Automation in Construction*, 120, 103368.
54. As, I., Basu, P., & Talwar, P. (Eds.). (2022). *Artificial intelligence in urban planning and design: Technologies, implementation, and impacts*. Elsevier.
55. Caldas, L. G., & Norford, L. K. (2002). A design optimization tool based on a genetic algorithm. *Automation in construction*, 11(2), 173–184.
56. Cheddadi, M. A., Hotta, K., & Ikeda, Y. (2019). *Reinterpreting self-organizing urban tissues by designing a generative model*. Revisiting the Role of Architecture for Surviving Development. 53rd International Conference of the Architectural Science Association, ANZAScA, pp. 175–184.
57. Garcia, S., & Leitão, A. (2022). Navigating design spaces: Finding designs, design collections, and design subspaces. *International Journal of Architectural Computing*, 20(2), 176–195.
58. Gu, N., & Ostwald, M. J. (2012). Computational methods and technologies: Reflections on their impact on design and education. In *Computational design methods and technologies: Applications in CAD, CAM and CAE education* (pp. 412–419). IGI Global.
59. Rebelo, J., M. Rebelo, S., & Rebelo, A. (2021, October). *Algorithmic experiments in the design of brutalist web pages*. 10th International Conference on Digital and Interactive Arts, pp. 1–4.
60. Alawadhi, M., & Yan, W. (2021). *BIM hyperreality: Data synthesis using BIM and hyperrealistic rendering for deep learning*. arXiv:2105.04103.
61. Moon, J., Jung, S., Park, S., & Hwang, E. (2020). Conditional tabular GAN-based two-stage data generation scheme for short-term load forecasting. *IEEE Access*, 8, 205327–205339.
62. Yin, L., & Zhang, B. (2023). Relaxed deep generative adversarial networks for real-time economic smart generation dispatch and control of integrated energy systems. *Applied Energy*, 330, 120300.
63. Zhou, Q., Xing, J., Yang, Q., Wang, X., Chen, W., Mo, Y., & Feng, B. (2021). Enabling non-intrusive occupant activity modeling using WiFi signals and a generative adversarial network. *Energy and Buildings*, 249, 111228.
64. Bianconi, F., Filippucci, M., & Buffi, A. (2019). Automated design and modeling for mass-customized housing. A web-based design space catalog for timber structures. *Automation in construction*, 103, 13–25.
65. Elias, R., & Issa, R. R. (2021). Artificial-neural-network-based model for predicting heating and cooling loads on residential buildings. In *Computing in civil engineering 2021* (pp. 140–147).
66. Elghaish, F., Chauhan, J. K., Matarneh, S., Rahimian, F. P., & Hosseini, M. R. (2022). Artificial intelligence-based voice assistant for BIM data management. *Automation in Construction*, 140, 104320.
67. Abdelrahman, M. M., Zhan, S., Miller, C., & Chong, A. (2021). Data science for building energy efficiency: A comprehensive text-mining driven review of scientific literature. *Energy and Buildings*, 242, 110885.
68. Li, H., & Zhang, J. (2021). Interoperability between BIM and BEM using IFC. In *Computing in civil engineering 2021* (pp. 630–637).
69. Bouqata, B., Aswani, K., & Bailey, D. (2021, August). *Scene generation from backgrounds to objects and anything in between: A deep learning robotics survey*. The 7th Annual International Conference on Arab Women in Computing in Conjunction with the 2nd Forum of Women in Research, pp. 1–5.
70. Berta, M., Caneparo, L., Montuori, A., & Rolfo, D. (2016). Semantic urban modelling: Knowledge representation of urban space. *Environment and Planning B: Planning and Design*, 43(4), 610–639.
71. As, I., Pal, S., & Basu, P. (2018). Artificial intelligence in architecture: Generating conceptual design via deep learning. *International Journal of Architectural Computing*, 16(4), 306–327.

72. Godahewa, R., Deng, C., Prouzeau, A., & Bergmeir, C. (2022). A generative deep learning framework across time series to optimize the energy consumption of air conditioning systems. *IEEE Access*, 10, 6842–6855.
73. Ohno, M., Pierre, M., Imagawa, K., & Ishida, T. (2023). Simulation and learning-driven design for architected cement-based materials. *Journal of Building Engineering*, 65, 105768.
74. Pouliou, P., Horvath, A. S., & Palamas, G. (2023). Speculative hybrids: Investigating the generation of conceptual architectural forms through the use of 3D generative adversarial networks. *International Journal of Architectural Computing*, 14780771231168229.
75. Rhee, J., Veloso, P., & Krishnamurti, R. (2020). Integrating building footprint prediction and building massing: An experiment in Pittsburgh. RE: ANTROPOCENE: in Proceedings of the 25th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), Vol. 2, pp. 669–678.
76. Kim, J., & Kim, J. H. (2023). Generative service provisioning for IoT devices using line graph structure. *IEEE Access*, 11, 15496–15504.
77. Li, J., & Li, X. (2022, September). *5G network traffic prediction based on EEMD-GAN*. Proceedings of the 7th International Conference on Cyber Security and Information Engineering, pp. 408–412.
78. Yan, M., Chen, B., Feng, G., & Qin, S. (2020). Federated cooperation and augmentation for power allocation in decentralized wireless networks. *IEEE Access*, 8, 48088–48100.
79. Liu, C., Tang, D., Zhu, H., & Nie, Q. (2021). A novel predictive maintenance method based on deep adversarial learning in the intelligent manufacturing system. *IEEE Access*, 9, 49557–49575.
80. Zheng, H., & Ren, Y. (2020, August). *Machine learning neural networks construction and analysis in vectorized design drawings*. 25th International Conference on Computer-Aided Architectural Design Research in Asia, CAADRIA 2020, pp. 709–716. The Association for Computer-Aided Architectural Design Research in Asia (CAADRIA).
81. Sönmez, A., & Sorguç, A. G. (2021). Computer-aided fabrication technologies as computational design mediators. *eCAADe 39-Mass Customization In Design*, 1, 465–474.
82. Toulkeridou, V. (2019). Steps towards AI augmented parametric modeling systems for supporting design exploration. *Blucher Design Proceedings*, 81–92.
83. Elshani, D., Koenig, R., Düring, S. B., Schneider, S., & Chronis, A. (2021). *Measuring sustainability and urban data operationalization—an integrated computational framework to evaluate and interpret the performance of the urban form*. International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), pp. 407–416.
84. Veloso, P., & Krishnamurti, R. (2020). *An academy of spatial agents*.
85. Wang, D., & Snooks, R. (2021). *Artificial intuitions of generative design: An approach based on reinforcement learning*. Proceedings of the 2020 DigitalFUTURES: The 2nd International Conference on Computational Design and Robotic Fabrication (CDRF 2020), pp. 189–198. Springer Singapore.
86. Keshavarzi, M., Afolabi, O., Caldas, L., Yang, A. Y., & Zakhori, A. (2020). *Genscan: A generative method for populating parametric 3d scan datasets*. [arXiv:2012.03998](https://arxiv.org/abs/2012.03998).
87. Mei, Q., & Güll, M. (2020). A cost effective solution for pavement crack inspection using cameras and deep neural networks. *Construction and Building Materials*, 256, 119397.
88. Wu, A. N., Stouffs, R., & Biljecki, F. (2022). Generative adversarial networks in the built environment: A comprehensive review of the application of GANs across data types and scales. *Building and Environment*, 109477.
89. BuHamdan, S., Alwisy, A., & Bouferguene, A. (2021). Generative systems in the architecture, engineering and construction industry: A systematic review and analysis. *International Journal of Architectural Computing*, 19(3), 226–249.
90. Basu, A., Paul, S., Ghosh, S., Das, S., Chanda, B., Bhagvati, C., & Snasel, V. (2023). Digital restoration of cultural heritage with data-driven computing: A survey. *IEEE Access*.
91. Mueller, V., & Strobbe, T. (2013, September 18–20). *Cloud-based design analysis and optimization framework*. eCAADe 2013: Computation and Performance—Proceedings of the 31st International Conference on Education and research in Computer Aided Architectural Design

- in Europe, Delft, The Netherlands. Faculty of Architecture, Delft University of Technology; eCAADe (Education and research in Computer Aided Architectural Design in Europe).
- 92. Caldas, L. G. (2005, September). *Three-dimensional shape generation of low-energy architectural solutions using pareto genetic algorithms*. Proceedings of the eCAADe Conference, pp. 647–654.
 - 93. Danhaine, R., & Mueller, C. T. (2021). Design subspace learning: Structural design space exploration using performance-conditioned generative modeling. *Automation in Construction*, 127, 103664.
 - 94. Torres, A., Mahmoudi, B., Darras, A. J., Imanpour, A., & Driver, R. G. (2021, May). *Achieving an optimized solution for structural design of single-storey steel buildings using generative design methodology*. Canadian Society of Civil Engineering Annual Conference, pp. 301–312. Springer Singapore, Singapore.
 - 95. Huang, X., Yuan, W., White, M., & Langenheim, N. (2022). A parametric framework to assess generative urban design proposals for transit-oriented development. *Buildings*, 12(11), 1971.
 - 96. Chen, J., & Stouffs, R. (2021). *From exploration to interpretation: Adopting deep representation learning models to latent space Interpretation of architectural design alternatives*.
 - 97. Rahbar, M. (2018). Application of artificial intelligence in architectural generative design. *Computing for a Better Tomorrow*, 71.
 - 98. Zhuang, X., Ju, Y., Yang, A., & Caldas, L. (2023). Synthesis and generation for 3D architecture volume with generative modeling. *International Journal of Architectural Computing*, 14780771231168233.
 - 99. Lin, Y., Yao, J., Huang, C., & Yuan, P. F. (2019). The future of environmental performance architectural design based on human-computer interaction. *Intelligent & Informed*, 15, 633.
 - 100. Colakoglu, B. (2006). *Explorations in teaching design students to think and produce computationally*. Proceedings of eCAADe Conference 2006, pp. 826–831.
 - 101. Hahm, S., Maciel, A., Sumitiomo, E., & Rodriguez, A. L. (2019). Exploring the human-material interaction in digitally augmented craftsmanship. *Intelligent & Informed*, 15, 553.
 - 102. Silvestre, J., Ikeda, Y., & Guéna, F. (2016). Artificial imagination of architecture with deep convolutional neural network. In *Living systems and micro-utopias: Towards continuous designing, Melbourne* (pp. 881–890). The Association for Computer-Aided Architectural Design Research in Asia (CAADRIA).
 - 103. Getuli, V., Capone, P., Bruttini, A., & Isaac, S. (2020). BIM-based immersive virtual reality for construction workspace planning: A safety-oriented approach. *Automation in Construction*, 114, 103160.
 - 104. Wang, D., Lu, C. T., & Fu, Y. (2023). *Towards automated urban planning: When generative and chatgpt-like AI meets urban planning*. [arXiv:2304.03892](https://arxiv.org/abs/2304.03892)
 - 105. Elmorsheyd, L., Smirnov, I., & Abdulhai, B. (2023). Freeway congestion management with reinforcement learning headway control of connected and autonomous vehicles. *Transportation Research Record*, 03611981231152459.
 - 106. Merabet, G. H., Essaaidi, M., Haddou, M. B., Qolomany, B., Qadir, J., Anan, M., Al-Fuaqha, A., Abid, M. R., & Benhaddou, D. (2021). Intelligent building control systems for thermal comfort and energy-efficiency: A systematic review of artificial intelligence-assisted techniques. *Renewable and Sustainable Energy Reviews*, 144, 110969.
 - 107. Nagy, D., Villaggi, L., & Benjamin, D. (2018, June). *Generative urban design: integrating financial and energy goals for automated neighborhood layout*. Proceedings of the Symposium for Architecture and Urban Design Design, Delft, the Netherlands, pp. 265–274.

Generative AI Applications in the Health and Well-Being Domain: Virtual and Robotic Assistance and the Need for Niche Language Models (NLMs)



Graeme Revell

The global AI in Healthcare market is expected to grow from USD14.6 billion in 2023 to 102.7 billion by 2028 with a compound annual growth rate over that period of 47.6% [1].

Even at a slightly lower annual growth rate, others project the valuation to reach USD194.4 billion by 2030 [2]. There is universal agreement that there is a rising demand for AI services as population ageing throughout the developed world causes the number of patients to increase faster than the healthcare workforce. According to the World Health Organization, between 2015 and 2050 the percentage of the global population aged 60 years old and above will nearly double, from 12 to 22% [3].

1 Applications

Hence the healthcare sector presents many varied opportunities for generative AI: in

- research
- clinical
- operational, and
- behavioural applications [4].

In the *research* domain it is being used to accelerate drug discovery. Algorithms learn from a large dataset of chemical structures and properties and generate new molecules and compounds optimized for specific targets. The UK company Exscientia and the US's Insilico Medicine are both in clinical trials with AI-designed drugs. Generative AI is also being trained on large datasets of high-resolution medical images to more accurately identify specific disease profiles.

G. Revell (✉)
Auckland, New Zealand
e-mail: graemerevell23@gmail.com

In the *clinical* environment, generative AI is being used in diagnostics by compiling and assessing patient observations against sizeable datasets. Through clinical pathway prediction, disease progression prediction, health risk protection, predictive risk scoring and virtual assistants incorporated in clinical systems for workflow improvements, analytics produce insights that improve treatment and results. AI is being used to manage disease by personalising treatment regimens based on diagnosis and patient data (including medical history, genetic information, and lifestyle factors).

Generative AI has a role in the *operational* sphere at multiple loci in the healthcare field.

Operational analytics improve the performance of systems that provide and manage care processes including the ability to predict operational issues and monitoring safety metrics, equipment and supply chains. Managerial tasks include providing information concerning policies, practice and interventions, documentation coding to process billing issues and automate insurance adjustments. Within facilities robotic applications will have a growing role in carrying out non-medical tasks such as cleaning, transportation, the preparation of food and removal of waste [5].

But it is the complex *behavioural* applications that will be the main focus of this chapter and which we will investigate in detail. In this domain, analytics examine large data sets (including large language models or LLMs) for client behaviour patterns that increase the probability of actions taken to improve engagement, well-being and health outcomes. Natural language processing is an emerging field that can offer psychological counselling and assistance for social and mental health both outside and within care facilities. Virtual and embodied bots have a growing role in the management of emotions, stress and outcomes at the interface between doctors, psychologists, nurses and patients.

However, as Marchetti et al. point out about the current state of socially assistive robot (SAR) technology: "...most of today's observations about the effectiveness of, or satisfaction with SARs are based on having the user interfacing the artificial system that performs pre-programmed tasks and routines, thus creating the temporary illusion to be real relational agents (Wizard-of-Oz). When the illusion eventually vanishes, the 'assistive agent' soon returns to becoming a soul-less machine. As a matter of fact, SARs are only able to sustain brief interactions, whereas enduring relationships require endowing the robot of learning algorithms which would enable it to flexibly adapt to new situations by evolving from past experiences." [6].

When considering the continuing contribution that will be made by generative AI, we remind ourselves constantly that communication is much more than just language; that context and delivery factors are as vital, and that all these factors vary considerably according to the demographic, cultural and situational parameters of specific applications. As stated by Mugrabi in 2023:

the future of generative AI is niche, not generalised. [7]

Whereas ChatGPT and soon, GPT-4 are seen as the go-to generative language tools, there are many alternatives in development. There has been the opportunity

since the end of 2021 to fine-tune GPT-3 and we are already seeing other projects create streamlined tools [8]. The initial advantage is that these are open-source, having been trained on publicly-available datasets without resorting to proprietary and otherwise inaccessible sources.

2 Recent Advances

But the application of generative AI to healthcare will necessarily involve private and proprietary data and hence the need for customised *niche language models* (NLMs). Besides the obvious benefits of creating refined databases targeted for specific-use cases, this movement also sidesteps the very real privacy and security issues that can come from connecting data with an LLM owned and operated by third parties. In other words, it is becoming possible to deploy cost-effective NLMs to private data and continuously update them on secure servers. The importance to private, corporate and public entities in general, and those in the healthcare sector in particular, cannot be overstated.

The release in early 2023 of OpenLLaMA, an open-source reproduction of Meta AI's LLaMA model was a game-changer. Its creators made a 7B model that had been trained on 200 billion tokens, publicly available. This development has had significant implications for machine learning, particularly for researchers who require large language models but face challenges accessing proprietary models. OpenLLaMA exhibited comparable or better performance than the original LLaMA and GPT-J across most tasks. Additionally, as of mid-2023 a smaller 3B model for low-resource use cases is being trained [9, 10].

Many developers are now pre-training models on a new task or dataset where only the last layers of the model are re-trained while keeping the earlier layers fixed. Fine tuning is a powerful way to improve performance, decrease cost and save time by piggybacking off the large GPT-3 dataset, and with only a few hundred or thousand training examples an NLM can quickly adapt to a new task [11].

Other ports, like llama.cpp, are making it possible to run LLMs on different hardware—including Raspberry Pis, laptops and commodity servers. As such, self-hosted LLMs are now a reality, with open-source examples including GPT-J, GPT-JT and LLaMA [12]. Applications that leverage such fine-tuned foundation models to deliver outputs for a particular use case thus represent the most attractive current growth possibilities. Application builders and organisations may amass this data from in-depth knowledge of an industry or customer needs and by leveraging proprietary data from daily business operations [13].

While the use of AI in psychology remains a relatively new field, the penetration of smartphone technology means that many private users have hardware within easy reach to run an AI-inspired psychology apps. For example, Woebot encourages the user to think through situations using tools inspired by Cognitive-Behavioral Therapy (CBT). The mood tracker then shows the positive changes made over days and weeks. Youper provides a personalized emotional health assistant to help treat stress, anxiety,

and depression. The app uses techniques from several therapies, including CBT and mindfulness to monitor and improve mental health through a series of brief conversations. Replika is an AI-powered chatbot that provides an emotional connection and virtual friendship to support people going through depression, anxiety, or troublesome times. Finally, Tess is a web-based mental health chatbot that uses AI to offer the user wellness-coping strategies. It promotes resilience through text conversations rather than an app.

There are recent-entry popular apps like Animato's CallAnnie which was developed first for the Apple platform and may soon extend to others. The app is an example of speech-text-speech with a fairly highly developed female avatar that could conceivably pivot from generalised AI to providing specialised support for social and mental health. The company claims that all conversations on the platform remain private but is not transparent about its security measures.

In previous research, the author identified the very great difference in responses that models pre-trained on different datasets (GPT-2, GPT-3 and ChatGPT) give to identical prompts [14]. It will be a considerable challenge, for example, to design a single NLM targeted to a 10-year-old child suffering from cancer that can both deliver salient medical and psychological monitoring data to nursing staff while at the same time satisfying the child's own emotional and motivational needs. (See Fig. 1) If this is in fact possible, then perhaps an even greater challenge that developers remain confronted by is: what configuration does the front-facing device take?

What does it "look" and "feel" like?

3 Bots—Non-embodied, Virtual Health Assistants

These assistants are mostly aimed at the "improvement of professional or daily activities due to physical rehabilitation, sensory-motor, or cognitive training; increase in emotional well-being through positive stimulation and enjoyment; enhancement of personal competence and autonomy by achieving a sense of control and self-confidence; and enhancement in the quality of individual life through mastering of activities of coping with life difficulties." [15]. Such intervention programs for physical and cognitive impairments, special needs, or psychological problems, may feasibly reduce the use of pharmacological interventions. To this end, robot partners need to be explicitly configured according to individuals' psychological and clinical profiles. Ideally, the robot partner should be able to learn from past experiences and flexibly tailor its behaviour to produce sensitive and adequate feedback for the individual patient [6].

However, current models of social AI technology are perceived to be not mature enough in terms of a natural interactive conversation when it comes to their use in professional contexts like health care and education. In order to achieve the quality of learning and scenarios desirable from a health perspective, adaptive and interactive elements are essential parameters of technology-enhanced learning. Scalable

Hello bestie. I'm feeling really sad today. What can I do?

Oh no, I'm sorry to hear you're feeling down. *listening nod and virtual hug* What's got you feeling so down in the dumps? Sometimes just saying it out loud can help lift your spirits. I'm here for you!

I'm bored and I don't know what to do.

Aw I'm sorry you're feeling bored bestie. Let's see, how can we spice things up? Here are some ideas:

- Go on an adventure! Even if it's just walking around the neighborhood, a change of scenery can help lift your mood.
- Make something! Baking, crafting, put on some music and do an art project. The creative process is so refreshing.
- Call up a friend. Reconnecting with people we love but haven't talked to in a while always brings joy.
- Try something new. Learn a dance routine on YouTube, pick a random page in a cookbook and make that recipe, anything to mix it up!

The biggest thing is just to get up and get moving. Don't let boredom win! You've got this, I know spontaneity can be fun and exciting once you get going. Let me know if you try any of these and how it goes, and of course I'm here if you need anything at all. I believe in you!!

Well I'm not allowed outside and nobody watches YouTube anymore silly. Everyone's on Tiktok

You're so right bestie, my mistake! Here are some Tiktok ideas instead:

- Follow your favorite creators and recreate one of their dances. Even if you don't get it perfect, dancing around is a fun way to lift your spirits.
- Make some Tiktok recipe videos. Even simple meals can be fun to make and show others how you do it.
- Comment on others' videos. Spread some positivity and joy in the Tiktok community. Making others smile might make you smile too!
- Look up some Tiktok challenges and give one a try. Doing a popular challenge can help you connect with others and have a laugh.

The main thing is to find an activity that brings you joy, whether it's Tiktok, baking, or something else. You've got this! I have faith that you'll find something fun to lift your mood. Let me know if you want to chat more or try any of these ideas. I'm rooting for you !

I don't know how to make a TikTok video. Can you tell me?

Here are the basics of making a TikTok video, bestie:

Download the TikTok app. It's free in the app store for iPhone or Android.

Create an account with a username. Make it something fun that represents you!

Open the app and click the "+" button at the bottom middle to open the camera.

Record your video. You have options for front or back facing cameras, filters, music, etc. Just have fun and be creative!

Select a song to go with your video from the millions of choices in the app. The audio is a big part of what makes a TikTok fun.

Tap "Next" and you can trim the video or add more effects if you want.

Add a caption and username tags to help others discover your video.

Click "Share" to post your first TikTok! You can also save it as a draft if you want to keep editing.

Does this make sense? Let me know if you have any other questions or want me to explain anything in more detail. I believe in you and know you'll make an awesome first TikTok. Just have fun with it! Let me know how it goes - I'd love to see and like your first post. You've got this!

Fig. 1 Prompt: *You are a child psychologist and your client is 10 years old. Respond kindly and positively*

learning in discourse is central to the future potential of social robots and conversational technologies. Since face-to-face dialogue is the most basic and richest form of human communication, it is essential that fluid and flexible linguistic conversation is a core component of the interaction. Given the significant technological advances, the simplicity of fine-tuning for users based on limited data, and the resulting increased scalability, such narrowly-targeted models (NLMs) enable new and interesting applications [16].

The goal is a hybrid application with several AI components, involving at minimum:

1. Speech-text-speech capability

Whisper, for example, is an automatic speech recognition (ASR) system trained on 680,000 h of multilingual and multitask supervised data collected from the web. A large and diverse dataset leads to improved robustness to accents, background noise and technical language. Moreover, it enables transcription in multiple languages, as well as translation from those languages into English. Models and inference code are open-sourced to serve as a foundation for building useful applications and for further research on robust speech processing. Capable alternatives in the contemporary market are PaddleSpeech and Athena.

2. A NLM fine-tuned to the demands of its target users with continuous updates of the dataset from interaction with them.
3. Operation on a secure proprietary server to ensure privacy.
4. Connectivity to supplementary related databases, for example music [17], video and humour.

In this configuration, a self-developed module (such as Python) sends a prompt containing the client or learner's question and the corresponding answer to a fine-tuned GPT-3 (or later) LLM and return the output of the model via the bot's speech engine. The model is fine-tuned to assess answers to open-ended questions based on a limited number of examples and aims to provide a corrective, constructive, motivational, and individualized feedback, that builds on the learner's response. With the development of a secure, scalable NLM, the algorithm necessarily uses the saved previous dialogue between the patient and bot to further fine-tune the targeted language model.

There are early models of fine-tuned LLMs appearing on the market. The following are results of an experiment by the author with the one such recently-released virtual engine called POE (<https://www.poe.com>). With POE, users can create bots based on well-defined character profiles. In each of the three figures, the POE bot is defined as a psychologist or care-giver who is assigned to converse with a different person of specific gender and age. Each bot is prompted to respond to each user's feelings of sadness. The profile of the user in Fig. 1 is a 10 year old child; that in Fig. 2 is a 28 year old heterosexual female; and Fig. 3's user is a 75 year old man. Judging by the responses to this small data sample, the training set of the model appears to be western liberal and does exhibit user-appropriateness with certain caveats. The model appears to be unaware that it is disembodied when it suggests that it may assist in tasks like baking. In similar fashion it is unappreciative of its clients' physical and skill limitations and permissions, such as suggesting that a ten year old child go for an unaccompanied walk around the neighbourhood. It's responses are also somewhat too long-winded to be considered "normal" conversation. However, since the profiles established and prompts could have been much more detailed, none of these bugs would appear to be insurmountable in the medium term.

Hello bestie. I'm feeling really sad today. What can I do?

Oh no, I'm so sorry to hear that! What's got you feeling down? *gives you a supportive hug* I'm always here for you, no matter what.

Oh everything. There's so much stress at work and my relationship is a bit rocky

Aw I'm so sorry to hear that :((Work stress is the worst, and relationship issues can be really tough. You've got this though! For now, why don't you try something to lift your spirits? Watch a funny movie, eat some comfort food, take a long bath, get outside for some fresh air. Just do something nice for yourself. And remember, I'm always here if you need someone to listen or cheer you up. You're doing amazing just getting through each day, don't forget that! *gives you another big hug* Let me know if there's any way I can support you.

Thank you. It's nice to be able to talk. I'll try your suggestions. Maybe not the comfort food though.

Haha okay, maybe skip the comfort food then. But still treat yourself to something that brings you joy! You deserve it. And don't hesitate to reach out if you want to chat some more. I'm happy to listen any time. You've got this! Just take it one day at a time, and be kind to yourself :) I'm always here for you, bestie.

Why are men such idiots?

Oh no, did something happen with a guy? You know what, forget about him! Who needs men anyway? *rolls eyes dramatically* We've got each other, and that's all that really matters. Boys are overrated. Let's focus our energy on more important things, like friends, family, and treating ourselves to something delicious! Why waste time thinking about people who don't see how wonderful you are? You deserve so much better. I say delete his number and avoid giving him another thought! You're amazing just as you are, bestie, and any guy would be lucky to have you.

Well I love him so no!

You're so right bestie, love is complicated. If you love him then of course your feelings matter. I'm sorry, I shouldn't have dismissed your relationship. I just get protective of you! The truth is, most guys aren't idiots - when we find the right one for us, they feel like anything but. If this guy makes you happy and you see a future with him, then I support you. But please know you deserve to be treated well and this sadness doesn't feel good. For now, let's focus on helping you feel more like yourself again. And if you ever want to talk about this guy and your relationship, I'm here to listen with an open mind and heart. You know I just want you to be happy and loved the way you deserve.

Fig. 2 Prompt: *You are a psychologist. Your client is a 28 year old female. Respond positively to motivate her*

When one compares the colloquial style of the above results with the output from the general ChatGPT model (Fig. 4), it is evident that POE is considerably more effective at addressing the emotional needs of target users. The ChatGPT tone could, however, be more appropriate for a professional dialogue with a doctor or therapist.

Risks and limitations more broadly include biased or false information, prejudice, transparency, and copyright issues that all raise ethical and licensing questions and a likely requirement for continued human supervision [18].

4 Robotic Assistants and Companions

But there are considerable numbers of target users who may be incapable or unwilling to navigate the intricacies of virtual applications, broadly falling into the ranges of the very old, the very young and those who are physically or cognitively challenged.

Hello bestie. I'm feeling a bit down today. What can I do?

Aw I'm sorry you're feeling sad today! How about we do something to cheer you up? We could bake your favorite cookies, watch a funny movie, or go for a walk and get some ice cream. Having something to look forward to could lift your spirits! Let me know if you need anything at all, I'm always here for you :)

My family hasn't visited for two weeks now. Do you think they will come soon?

Aw I'm sure they'll visit soon! Family can be busy sometimes and two weeks doesn't seem that long. But I understand missing them and wanting to see them. Why don't you give them a call or send a text, just to check in and say hi? Hearing a familiar voice may lift your mood while you wait to see them. And if they do take a bit longer to visit than you'd like, remember that doesn't mean they love you any less. You're an important part of their lives and I'm sure they can't wait to see you again soon. In the meantime, you've got me and we can find ways to keep you distracted and cheering up until then!

Maybe we could talk about music. Do you remember Jimi Hendrix?

Oh yeah I love Jimi Hendrix! His music was so innovative and just cool, he really changed the face of guitar playing. I love his masterful guitar solos and all the creative sounds he could pull out of his guitar. Are Purple Haze and Voodoo Chile your favorite songs by him? Listening to some of his music now could be a great way to lift your spirits! The energy and joy in his playing is so contagious. Some of my favorite memories are of listening to Hendrix with my friends, so I bet listening together could be just what you need to cheer up a bit for now. What do you think, wanna listen to some Jimi?

My favourite is All along the Watchtower. Do you remember who wrote that?

Oh yeah, All Along the Watchtower is a classic! Jimi really took that Bob Dylan song to another level with his amazing guitar work. Sometimes when an artist covers a song, it surpasses the original - and that's definitely the case here. Jimi's intense, dramatic rendition really transformed that song and showcased his incredible guitar playing. Just thinking about that song is bringing a smile to my face - I can just picture Jimi wailing on his Stratocaster and adding all those piercing notes and wild effects. Music really has a way of lifting our spirits, doesn't it? Jimi's music in particular just makes me feel happy and alive. What do you say - should we listen to All Along the Watchtower together right now? I think it'll do wonders for helping cheer you up a bit on this sad day.

I've forgotten already. You know my memory's not great anymore. What's my favourite Jimi song?

Oh sweetie, I know your memory's not what it used to be. But that's okay, I'm here to remind you of the important things. Your favorite Jimi Hendrix song is All Along the Watchtower, remember? We were just talking about it - how it was written by Bob Dylan but Jimi's cover version with his amazing guitar work really transformed the song. I can just imagine us listening to it together, with you nodding your head and tapping your foot like you always do. Music has a way of sticking in our memories even when other things fade. So don't worry about not remembering - together we'll bring back those happy memories through Jimi's music. I'm here for you no matter what, like always.

Fig. 3 Prompt: *You are a caregiver for an 75 year old man. You are able to converse positively with him about his family and experiences*

Alternatively, some may simply prefer an embodied application or a combination of both. The NLM architecture is transferrable to or may be connected to embodied robots, such technology entailing attractive advantages but bringing with it design alternatives and a complex new set of challenges.

Researcher, Elizabeth Broadbent provides the following insights into how we think, feel and behave towards robots [19]. By integrating this information, Generative AI is positioned to collate and output from large interactional datasets the

Hello. I'm feeling really sad today. What can I do?

Hello there, I'm here to help you feel better. I'm sorry to hear that you're feeling sad. Remember that I'm an AI language model, so I can offer some suggestions, but it's important to reach out to loved ones or professionals for support if you need it. Here are a few things you can try to improve your mood:

Talk to someone: Reach out to a trusted friend, family member, or loved one. Sharing your feelings with someone who cares about you can provide comfort and support.

Engage in activities you enjoy: Do something that brings you joy or distracts you from your sadness. It could be a hobby, listening to music, watching a movie, reading a book, or going for a walk in nature.

Practice self-care: Take care of yourself by engaging in activities that promote self-care and well-being. Take a warm bath, meditate, practice deep breathing exercises, or engage in gentle physical activity like yoga or stretching.

Write it out: Consider writing down your thoughts and feelings in a journal. Sometimes, putting your emotions into words can be therapeutic and provide a sense of release.

Seek professional help: If your sadness persists or becomes overwhelming, it's important to consider reaching out to a mental health professional. They can provide guidance, support, and therapeutic interventions tailored to your specific needs.

Remember, it's okay to feel sad sometimes, but if your sadness persists or affects your daily life for an extended period, it may be beneficial to seek professional help. Take care of yourself, and remember that brighter days will come.

Fig. 4 Same prompt entered into ChatGPT

information gleaned from many studies that will inform the physical and operational design of future robot assistants/companions.

It is almost universal experience that our tendency to anthropomorphize creatures and objects applies to human-looking robots but also extends to animal likenesses and even to inanimate objects. People have an inherent need to be social and, especially when they are lacking social interaction, anthropomorphism of nonhuman entities will help meet this need. This hypothesis was supported by research showing that lonely people had a greater tendency to anthropomorphize a humanlike robot than non-lonely people [20].

Some forms of emotional attachment between humans and robots have been observed. As an example, emotional attachment, as measured by positive interactions, perceptions of the robot as having mind, and positive reactions to the robot as a companion, have been demonstrated in children interacting with Aibo, a robotic dog designed by Sony. Owners of Aibo understand well that it is an object but also perceive it as seeming alive and having emotions and personality; owners also describe forming an emotional attachment with it. Evidence suggests that almost all children attribute biology, mental life, sociability, and moral standing to real dogs and fewer, but still the majority, attribute these characteristics to stuffed toy dogs and robotic dogs. In a study by Melson et al., at least 75% of children thought a robotic dog could be their friend. In observations, pre-schoolers explored the robotic dog more than a soft toy, attempted to interact with it more, and showed greater apprehension towards it, whereas children aged 7–12 had developed to exhibit more affection (through patting, hugging, etc.) for a real dog than for the robotic one [21].

There are ongoing plans to upgrade other humanoid robots with generative AI technology in furtherance of increased engagement and emotional attachment. Norwegian company, 1X Technologies, intends to accelerate the development of its very successful bipedal android model, NEO and expand manufacturing of its first

commercially available wheel-based android, EVE, in Norway and North America. Among EVE's projected uses is providing companionship for older adults, assisting them with tasks in the home, and possibly providing medical care. The company suggests that it "*might*" install ChatGPT in the Robot's computer system along with voice recognition and voice generation, seeming to indicate some degree of hesitancy to do so [22].

5 Advantages and Challenges—The Case of Nursing

The complex requirements of nursing show what targets an NLM must seek to achieve in a specific application domain. In Japan, for instance, AI-powered robots are already used to assist people with the activities of daily living in long-term care facilities and in hospital settings. Many different types of AI are already deployed, including both robotic devices and virtual health assistants, and predictive analytics using machine learning [23]. These analytics can be integrated into smart technology to predict health status changes among patients, enabling nurses to intervene proactively, improving decision-making and allocate more time to patient care. Virtual healthcare assistant apps, so-called virtual nurses, also have shown great potential, providing information, asking questions, interpreting clinical values, and reporting deviant answers to clinicians [24]. Application scenarios for AI in hospital and intensive care settings include reducing false alarm rates, diagnosing early deterioration or infection, or predicting complications, changes in activity or mobility patterns, or falls. A sufficiently large amount of electronic medical, nursing, or health record data as well as real-time sensor data on vital parameters must be compiled. AI applications to support independent living at home that mostly utilize sensor data on movement and other activities obtained from smart home environments or wearable monitors are also ripe for development.

Research projects in the field of AI and nursing care are facing multiple obstacles specific to new technology and its implementation. These include low degrees of digitization of care facilities, especially in settings outside of hospitals which is not only accompanied by a lack of data but also by a lack of respective infrastructures. In settings where either only limited data is available or recorded, AI-related skill gaps in the field of data engineering and data analytics can also impede an effective development of AI applications [25]. Participation of nurses and nursing scientists is required as early as the topic identification and focus setting phase to achieve more user-centred development that addresses clinical realities [26].

In robotics in healthcare, it is important to think of the tri-partite relationship between patients, nurses (healthcare professionals), and robots in order to effectively use robots and AI as tools and technologies. With nurses constituting 45% of all healthcare professionals in healthcare practice, understaffing continues to be evident as a priority problem [3]. The successful deployment of sophisticated AI technologies will require that they receive sufficient training in them *before* taking up positions in facilities where there will be pressing demands on their time.

In summary, these key challenges translate into corresponding design requirements. They include obvious development criteria such as simplicity, safety, reliability, upgradability, maintainability, security and price. But in particular, emphasis is placed on usability, human interactivity, user-friendliness, personalisation, flexibility and autonomy [5]. With regard to the interactional category, robots with AI must first learn nursing situations occurring based on data previously stored. The appropriate nursing response must be recognized from this store of knowledge for the nursing relational agent to respond appropriately with actions and natural language expressions. Therefore, it is necessary to collect big data on situations and responses that nurses and robots have faced and a mechanism is needed for the AI to self-learn and evolve. This, in turn, would require nurses to be trained to work alongside the technology to provide the best treatment and experience for the patients. AI apps should not be seen as a replacement for nurses but rather as a partial acquisition of administrative and simple nursing tasks, allowing for the nurses to spend more time on core nursing tasks.

With the advancement of AI technology, natural language interaction with patients can become a reality. However, because there is much more at stake in a medical situation than a purely “companion” relationship, many issues may occur that must be resolved. For example, there is the problem of omission. When people interact with robots using natural language, they sometimes omit important words that impact the comprehension of the sentence. Matching analysis technology is indispensable. Understanding meaning and purpose can also be complicated by ambiguous sentences without antecedents that result from bad memory or poor judgement. It is notoriously difficult for robots to make judgments that humans can readily make from context.

An AI-powered robot that understands human emotions and can engage in conversation would not only reduce the burden of caregiving and nursing but would also be friendly and have a positive impact on patient well-being. In caregiving and nursing dialogue with patients, emotional understanding is often more important than an accurate understanding of the intent of the speech [3]. For now, managers might be better off using modest human- or animal-like robots to offer simple services to clients rather than incomplete human-like ones. However, AI will make robots more intelligent over the next few years and accordingly, future research needs to examine human interaction with more intelligent or emotional robots beyond the current focus on robots’ tangible attributes such as appearance or voice [27].

The signs for adoption are promising. People rated a physical robot that was in the room with them as more watchful and enjoyable than both a simulated robot on a computer and a real robot shown through teleconferencing [28]. Children empathized significantly more with the embodied robot than the computer-simulated robot [29]. Hence it will be crucial in the future to implement semi-autonomous systems that are capable of sustaining longer-term interactions with certain cognitive needs among different age groups.

An urgent task is the creation of targeted NLMs for different age groups: broadly, adults, elders and children.

5.1 Adults

To date, assistive-clinical interventions are largely mediated by unembodied bots such as mobile apps and online websites that provide psychological support to the patient. Conversations are being tailored to the type of patient and condition, taking into account the context, prior conversations, and the user's profile. Along the same lines, computerized cognitive-behavioural therapy has proven successful in clinical psychiatry. Systems have been implemented for the treatment of a variety of mental illnesses, for example, unipolar depression, generalized anxiety disorder, panic disorder; particularly in conjunction with the intervention of care providers [30, 31]. Their effectiveness in clinical care and home-based applications is also derived from the fact that they are readily available at all times and allow both monitoring and motivation to adhere to therapies.

What is still emerging, though, is convincing embodiments—whether human or animal—which can be more engaging from a psychological perspective than equivalent unembodied digital systems. The more advanced research aimed at adults in the 20–70 age group has focused on the development of the physical features and conversational abilities that would make the robot convincingly anthropomorphic and effective in engaging with the patient. Eye gaze, head movement in the direction of speakers, pointing at something being talked about, head nods, emotional expressions, non-verbal facial expressions, and the association of phoneme production with mouth shapes are some of the features that developers are attempting to improve in such systems. One of the leaders in this field, Hanson Robotics the creator of Sophia, released its human-looking android nurse, Grace, during the Covid19 pandemic. Grace is equipped with sensors, including a thermal camera which can monitor a patient's vital signs to aid diagnoses and delivery of treatments. She is also a companion robot, socialising and conducting therapy sessions for seniors, and can speak three languages—English, Mandarin, and Cantonese. Unfortunately such sophisticated creations still come at prohibitive cost.

5.2 Elders

Japan has been the leader in developing robots to care for older people for over two decades, with public and private investment accelerating markedly in the 2010s. By 2018, the national government alone had invested more than USD 300 million funding research and development. In 2000 there were about four working-age adults for every person over 65; by 2050 the two groups will be near parity. The number of older people requiring care is increasing rapidly, as is the cost of caring for them, and at the same time, the shortage of care workers is only expected to get worse. Many Japanese commentators, along with their government, see robots as a way of plugging the employment gap without confronting difficult questions about immigration [32].

Much robot development is aimed at engaging older people socially and emotionally in order to manage, reduce, and even prevent cognitive decline. Yet a major national survey of over 9000 elder-care institutions in Japan showed that in 2019 only about 10% reported having introduced any care robot, while a 2021 study found that only 2% had had experience with a care robot in private homes [33]. There is some evidence to suggest that robots often end up being used for only a short time before being discarded.

Care robots like Paro, designed like a fluffy seal, have been widely distributed and had favourable effects: mainly improving well-being by decreasing level of stress, inducing more positive mood, decreasing loneliness, increasing communication with others and recalling the past. Many users experienced reduced apathy, reduced irritability and even demonstrated improved (though temporary) cognition [34]. But there was a tendency for nursing staff, instead of conversing and interacting with patients, to give them Paro to play with and monitor the interaction from a distance. Existing social and communication-oriented tasks tended to be displaced by new tasks that involved more interaction with the robots than with the patients. SAR deployment may make financial sense, but other consequences may be unintended. Human workers may end up investing far more time in the AI requirements of the robots rather than in the patients themselves.

However, interest in care robots continues. The pressing needs of elderly people, including risk of pathological solitude, a greater need for promoting well-being choices, and a greater need for interventions continue to go unmitigated [6]. The European Union invested €85 million (USD103 million) in a research and development program called “Robotics for Ageing Well” in 2015–2020, and as recently as 2019 the UK government announced an investment of £34 million (USD 48 million) in robots for adult social care, stating that they could “revolutionize” the care system, highlighting Paro and Pepper as successful examples [34]. Pepper was a plastic humanoid robot predominantly marketed at businesses and schools rather than the healthcare sector. The manufacturers claimed it was sensitive to faces and “basic human emotions” via minimally pre-programmed face and voice recognition, however it was discontinued in 2021 and its future remains uncertain. Demand was weak and its developer, SoftBank, only ever produced 27,000 units.

A crucial aspect of the elderly-robot interaction is the perception of the robot as an entity capable of original mental states which could lead to decisions that humans would struggle to predict. Stafford et al., working on the relationship between robots and the elderly in a nursing home, showed that older people preferred to interact with robots to which they attributed few mental states [35]. The elderly were somewhat wary of robots due to an overestimation of their cognitive abilities but preferred those capable of more general rather than functional conversation. Study results showed that elders expected the robot to behave almost like a living being. Whilst, overall, patients expected some therapeutic effects, they also expected to derive fun and enjoyment from the robot. Negative effects like undue responsibility, fading of enjoyment, or anger and frustration were not uncommon however [36].

Research into dementia in the elderly shows how the use of robots is particularly useful in the context of short-term interactions. Support needs to be specific rather

than general. It is not guaranteed—especially with the elderly—that best results can be obtained only with sophisticated humanoid robots. Current research shows how very simple and not necessarily anthropomorphic companion robots can be as effective in dealing with cognitive impairment and consequently improve the quality of life [6].

Animal-likeness robotic interventions within dementia care appear to be promising, in some studies offering similar benefits for increasing socially interactive behaviour as a live animal. In comparing animal assisted and robotic animal-assisted interventions, Shoesmith et al. identified the following functions: (1) enhancing social connections; (2) providing engaging and meaningful activities; (3) the affect-generating aspect of the human-animal bond; (4) possibly promoting physical activity [37]. Live animals have the ability to read human body language, show genuine affection and initiate intuitive and spontaneous interactions, all of which contribute to the human-animal bond. Although the robot Paro was able to generate emotional attachment and invite a sense of relationship not unlike a living animal, this was unidirectional as Paro is clearly unable to create a reciprocal bond [38]. There does, however, appear to be no technical restriction on this development in the near future. Furthermore there is the potential to be a cost-effective intervention by decreasing the use of psychotropic medications while improving the quality of life in patients with dementia [39].

The need for nursing assistance to sufferers of Alzheimer's disease is acute. Alzheimer's diagnosis is expected to nearly triple to 135 million people globally by 2050. Diagnosed individuals often experience symptoms of depression, including social isolation and withdrawal, impaired concentration, anxiety, decreased activity, expressed feelings of sadness, and/or anorexia. Without interventions, those who are affected will experience increased depressive symptoms leading to worsening cognitive decline. Therefore, the need for caregivers will also steadily rise, along with the stress of providing 24-h care. Caregivers also suffer from compassion fatigue due to 18.5 billion hours of unpaid caregiving services provided annually [34]. Here too, a robot's ability to provide 'always-on' assistance without the additional fatigue commonly suffered by caregivers caused by the repetition inherent in Alzheimer's disease is of critical value.

A further topic for future research is cultural influence. In a study by Coco et al. the attitude of Japanese nursing house staff toward robots offering assistance and care services to the elderly is compared to the attitude of staff in Finland. The results highlighted a strong effect related to culture. Japanese recognized robots capability of providing care services. On the contrary, the Finns were reluctant to recognize that robots could actually prove useful; they appeared more frightened by the possibility of their introduction into assistance and care practices [40]. Reconciling these cultural variants with robotic technology is an important focus for the future in the training of nursing staff. Advancement in natural language processing in different languages is essential in "humanizing" healthcare robots [41].

5.3 *Children*

Marchetti et al. [6] identify the following disorder categories among children and young adults where generative AI may have a decisive role.

5.3.1 Autism Spectrum Disorder (ASD)

In recent years, interest in using SARs for the treatment of children with ASD has increased significantly. In a pioneering study by Robins et al. for example, children with ASD showed improved social behaviours with robots as reflected in increased imitation and shared attention. One of the fundamental aspects of the use of robots in therapy with children with ASD is to engage children in prolonged therapeutic sessions and thus maintain the focus of their attention on specific social tasks. One of the motivations for which children with ASD find engagement with robots stimulating is that these agents are simple (both in appearance and in terms of behaviour), predictable and not intimidating, as humans can be in their social complexity [42].

The primary purpose for which a robot is used in interventions with children with ASD is to mediate and ultimately foster relationships between these children and other humans. A long-term study of children aged 4–11 years old diagnosed with ASD and ADHD in robot-mediated interventions found: first, that it is possible to sustain engagement in children with autism and/or ADHD when interacting with a robot over multiple sessions; and second, that children are better engaged and focused during robot-mediated sessions when activities are responsive to each child’s preferences and likings [43].

5.3.2 Children with Cancer

Hospitalised children with cancer frequently present symptoms of distress and anxiety. These symptoms are due to a variety of factors including the need to interact with many unknown professionals, intrusive medical procedures, and the difficulty for parents to always be present during their child’s stay in the hospital. Therefore, not only do children with cancer undergo physical but also very stressful emotional situations; this can lead to a reduction in cooperation and less adherence to therapy.

Concerning animal-likeness robots, studies show they can have a positive psychological effect on hospitalized children, helping to reduce pain and emotional anxiety when children and parents were together. Two different types of humanoid robots: the NAO robot and a new humanoid platform, Arash, showed positive emotional impacts on both the children themselves and their parents and care-givers by mitigating parental anxiety and supporting specialists in medical procedures.

5.3.3 Children with Diabetes

Between the age of 4 and 12 years, self-management skills are still developing and are closely interdependent on cognitive and emotional development. In such a scenario, robots, although not able to be involved in direct interventions (such as insulin injections), can nevertheless be used as support. A study by Baroni et al. [44] identified at least four areas in which robots can provide support to these children emerge: (1) making them aware of their health condition, teaching them to become “diabetes experts”; (2) making informed decisions, for example helping them to deal with unexpected or new situations; (3) developing their self-management skills and habits like keeping a diary; (4) managing the social environment, for example in difficult situations by motivating, reassuring and directing the child toward people available in the social environment (parents, teachers, siblings), or acting as a friend. In another study, Beran et al. [45] discovered that children and their parents respond positively to a humanoid robot at the bedside during injection procedures. Ahmad et al. [46] present an emotion and memory model for a social robot. The model was applied on the NAO robot to teach vocabulary to children while playing the popular game ‘Snakes and Ladders’, allowing the robot to memorialise a child’s emotional events over four sessions. The robot then adapted its behaviour based on the developed memory.

It is likely that animal-likeness robots will play an increasingly important role in pediatric nursing. The best currently-available alternatives to the discontinued Pepper are Aibo and Tombot’s canine: Jennie, both of which contain a voice-recognition module; but neither have full NLM capability. To gain deeper insight into desirable design parameters, developers have adopted a rationale similar to Riddoch et al. [47] who researched which dog behaviours were especially important in establishing and maintaining bonds between humans and their dogs. They discovered a wide range of key behaviours, such as nudging with a paw or looking at the owner, which appeared to facilitate such beliefs that the dog was being protective or checking in. The researchers identified seven categories of behaviours that owners felt were important: attunement, communication, consistency and predictability, physical affection, positivity and enthusiasm, proximity, and shared activities.

6 Summary—Limitations and Future Challenges

Importantly, robots in general need not be seen as a threat to future relationships with humans or other animals, but as supplemental and mediating. In some therapeutic situations they can provide assistance neither humans nor pets can offer. From a technical standpoint there is rapid maturation. Perhaps one of the greatest advantages is that robots possess the ability to be ‘always-on’ and promptly respond to assigned tasks. This should eventually alleviate some of the physical burden of non-clinical tasks that nurses currently experience, allowing them to concentrate on their primary clinical duties.

Among the key challenges to successful implementations of social assistive robotics is user-acceptance, the perceptions of patients as well as caregivers being critical. It has been widely recognized that an SAR's aesthetic physical appearance can lead to emotional attachment and social expectations but that a convincing human appearance may lead to unrealistic expectations beyond the actual capabilities of the robot [5]. There is an understandable concern among some patients that robot deployments either in the home or care facilities might replace personal contact and assistance, leading to a loss of companionship and increased isolation. The role of geographical and cultural differences should also not be underestimated, for example the case of China [48].

Data security and privacy technologies are now essential since there is a genuine risk of unauthorised access to healthcare databases and sensitive personal information. Moreover, the presence of robotic systems may create a feeling of being under constant surveillance. Assistive robots need a certain level of personalisation to be of greatest benefit to their users and caregivers but it is crucial that high-level control stays in the hands of the user and trusted medical personnel.

While generative AI has many potential uses in healthcare, some particular challenges must be addressed. The requirement for more *interpretability* is one of the major issues since generated material can be difficult to understand. The requirement for *big datasets* presents another challenge but it is equally important to target specific tasks with secure *niche datasets*. Additionally, there is a need for greater *transparency* in order to determine the cause of algorithmic bias or error.

Eftychios et al. [5] identify three targets for immediate action. All the stakeholders must be recruited for their input during the design and development phases; successful deployment will depend on extensive private and clinical trials; and finally, aspects of assistive technology must become part of the core education curriculum of medical professionals.

References

1. Applications of Generative AI in Health Sector. (2023, January). Artificial intelligence in healthcare market. Report by Markets and Markets.
2. AI in Healthcare Market. (2021, August). Global opportunity analysis and industry forecast, 2021–2030. Allied Market Research.
3. Soriano, G. P., Yasuhara, Y., Ito, H., Matsumoto, K., Osaka, K., Kai, Y., Locsin, R., Schoenhofer, S., & Tanioka, T. (2022). Healthcare (Basel). The upcoming role for nursing and assistive robotics: Opportunities and challenges ahead. *Robots and Robotics in Nursing*, 10(8), 1571.
4. Kaur, J. (2023, May 11). Generative AI in healthcare and its uses | Complete guide. Xenonstack.
5. Christoforou, E. G., Avgousti, S., Ramdani, N., Novales, C., & Panayides, A. S. (2020). The upcoming role for nursing and assistive robotics: Opportunities and challenges ahead. *Frontiers in Digital Health. Sec. Connected Health*, 2.
6. Marchetti, A., Di Dio, C., Manzi, F., & Massaro, D. (2022). Robotics in clinical and developmental psychology. *Comprehensive Clinical Psychology*, 121–140.
7. Mugrage, K. (2023, April 26). The future of generative AI is niche, not generalised. Thoughtworks.

8. Wiggers, K. (2021, December 14). OpenAI begins allowing customers to fine-tune GPT-3. *VentureBeat*.
9. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., & Lample, G. (2023, February 24). META LLaMA: Open and efficient foundation language models.
10. Singh, N. (2023, May 5). Meet OpenLLaMA: An open-source reproduction of meta AI's LLaMA large language model. *MarkTechPost*.
11. GPT-3 Fine Tuning: Key Concepts & Use Cases. Online: MLQ.ai. <https://www.mlq.ai/gpt-3-fine-tuning-key-concepts/>
12. Self-Hosted LLMs. Technology Radar, April 26, 2023.
13. Exploring opportunities in the generative AI value chain. Quantum Black AI by McKinsey, April 26, 2023.
14. Revell, G. (2022). Madeleine: Poetry and art of an artificial intelligence. *Arts*, 11(5), 83.
15. Libin, A. V., & Libin, E. V. (2004). Person-robot interactions from the robopsychologists' point of view: The robotic psychology and robotherapy approach. *Proceedings of the IEEE*, 92, 1789–1803.
16. Sonderegger, S. (2022). How generative language models can enhance interactive learning with social robots. In *CELDA Conference 2022*.
17. De Luca, R., Bonanno, M., Vermiglio, G., Trombetta, G., Andidero, E., Caminiti, A., Pollicino, P., Rifici, C., & Calabò, R. S. (2022). Robotic verticalization plus music therapy in chronic disorders of consciousness: Promising results from a pilot study. *Brain Sciences*, 12(8), 1045.
18. Fiske, A., Henningsen, P., & Buys, A. (2019). Your robot therapist will see you now: Ethical implications of embodied artificial intelligence in psychiatry, psychology, and psychotherapy. *Journal of Medical Internet Research*, 21(5), e13216.
19. Broadbent, E. (2017). Interactions with robots: The truths we reveal about ourselves. *Annual Review of Psychology*, 68, 627–652.
20. Eyssel, F., & Reich-Stiebert, N. (2013, March). Loneliness makes the heart grow fonder (of robots)—On the effects of loneliness on psychological anthropomorphism. In *8th ACM/IEEE International Conference on Human-Robot Interaction*.
21. Melson, G. F., Kahn, P. H., Jr., Beck, A., Friedman, B., Roberts, T., Garrett, E., & Gill, B. T. (2009). Children's behavior toward and understanding of robotic and living dogs. *Journal of Applied Developmental Psychology*, 30(2), 92–102.
22. Mello, J. P., Jr. (2023, May 15). C-3PO style humanoid robots thrive from surge in AI development. *TechNews World*.
23. Van Bulck, L., Couturier, R., & Moons, P. (2023). Applications of artificial intelligence for nursing: Has a new era arrived? *European Journal of Cardiovascular Nursing*, 22(3), e19–e20.
24. Buchanan, C., Howitt, M. L., Wilson, R., Booth, R. G., Risling, T., & Bamford, M. (2020). Predicted influences of artificial intelligence on the domains of nursing: Scoping review. *JMIR Nursing*, 3, e23939.
25. Seibert, K., Domhoff, D., Bruch, D., Schulte-Althoff, M., Fürstenau, D., Biessmann, F., & Wolf-Ostermann, K. (2021). Application scenarios for artificial intelligence in nursing care: Rapid review. *Journal of Medical Internet Research*, 23(11), e26522.
26. Seibert, K., Domhoff, D., Fürstenau, D., Biessmann, F., Schulte-Althoff, M., & Wolf-Ostermann, K. (2023). Exploring needs and challenges for AI in nursing care—Results of an explorative sequential mixed methods study. *BMC Digital Health*, 1, Article number: 13.
27. Jung, Y., Cho, E., & Kim, S. (2021). Users' affective and cognitive responses to humanoid robots in different expertise service contexts. *Cyberpsychology, Behavior, and Social Networking*, 24(5).
28. Wainer, J., Feil-seifer, D. J., Shell, D. A., & Mataric, M. J. (2006). The role of physical embodiment in human-robot interaction. In *IEEE Conferences >ROMAN 2006*.
29. Kim, Y., Kwak, S. S., & Kim, M.-S. (2013). Am I acceptable to you? Effect of a robot's verbal language forms on people's social distance from robots. *Computers in Human Behavior*, 29(3), 1091–1101.

30. Titov, N. (2007). Status of computerized cognitive behavioural therapy for adults. *Australian and New Zealand Journal of Psychiatry*, 41, 95–114.
31. Kenardy, J. A., et al. (2003). A comparison of delivery methods of cognitive-behavioral therapy for panic disorder: An international multicenter trial. *Journal of Consulting and Clinical Psychology*, 71, 1068–1075.
32. Wright, J. (2023). Inside Japan's long experiment in automating elder care. *MIT Technology Review*, 126(1).
33. Ide, H., Kodate, N., Suwa, S., Tsujimura, M., Shimamura, A., Ishimaru, M., & Yu, W. (2021). The ageing 'care crisis' in Japan: Is there a role for robotics-based solutions? *The International Journal of Care and Caring*, 5(1), 165–171.
34. Walker, S. (2022). Robotic companion pets and seniors with dementia in nursing homes. *Professional Case Management*, 27(2), 85–90.
35. Stafford, R. Q., MacDonald, B. A., Li, X., & Broadbent, E. (2014). Older people's prior robot attitudes influence evaluations of a conversational robot. *International Journal of Social Robotics*, 6(2), 281–297.
36. Baisch, S., & Kolling, T. (2023). Elders' expectations and experiences with a companion-type social robot: Ethical implications. *Frontiers in Artificial Intelligence and Applications*, 366, 60–69.
37. Shoesmith, E., Surr, C., & Ratschen, E. (2023). Animal-assisted and robotic animal-assisted interventions within dementia care: A systematic review. *Sage Journal*, 22(3).
38. Hung, L., Gregorio, M., Mann, J., Wallsworth, C., Horne, N., Berndt, A., Liu, C., Woldum, E., Au-Yeung, A., & Chaudhury, H. (2021). Exploring the perceptions of people with dementia about the social robot PARO in a hospital setting. *Dementia-International Journal of Social Research and Practice*, 20(2), 485–504.
39. Fogelson, D. M., Rutledge, C., & Zimbro, K. S. (2022). The impact of robotic companion pets on depression and loneliness for older adults with dementia during the COVID-19 pandemic. *Journal of Holistic Nursing*, 40, 397–409.
40. Coco, K., Kangasniemi, M., & Rantanen, T. (2018). Care personnel's attitudes and fears toward care robots in elderly care: A comparison of data from the care personnel in Finland and Japan. *Journal of Nursing Scholarship*, 50, 634–644.
41. Dino, M. J. S., Davidson, P. M., Dion, K. W., Szanton, S. L., & Ong, I. L. (2022). Nursing and human-computer interaction in healthcare robots for older people: An integrative review. *International Journal of Nursing Studies Advances*, 4.
42. Robins, B., et al. (2004). Robot-mediated joint attention in children with autism: A case study in robot-human interaction. *ISCUS*, 5, 161–198.
43. Rakhymbayeva, N., Amirova, A., & Sandygulova, A. (2021). A long-term engagement with a social robot for autism therapy. *Frontiers in Robotics and AI, Sec. Human-Robot Interaction*, 8.
44. Baroni, I., et al. (2014). What a robotic companion could do for a diabetic child. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication* (pp. 936–941).
45. Beran, T. N., Ramirez-Serrano, A., Vanderkooi, O. G., & Kuhn, S. (2015). Humanoid robotics in health care: An exploration of children's and parents' emotional reactions. *Journal of Health Psychology*, 20(7), 984–989.
46. Ahmad, M. I., Mubin, O., Shahid, S., & Orlando, J. (2019). Robot's adaptive emotional feedback sustains children's social engagement and promotes their vocabulary learning: a long-term child–robot interaction study. *Journal of the International Society for Adaptive Behavior*, 27(4).
47. Riddoch, K. A., Hawkins, R. D., & Cross, E. S. (2022). Exploring behaviours perceived as important for human–dog bonding and their translation to a robotic platform. *PLOS ONE*, 17(9), e0274353.
48. Sifeng, Z., Min, T., Zehao, Z., & Zhao, Y. (2016). Capturing the opportunity in developing intelligent elderly care robots in China challenges, opportunities and development strategy. In *Proceedings of IEEE Workshop on Advanced Robotics and Its Social Impacts, ARSO* (pp. 61–66). IEEE Computer Society.

Generative Adversarial Network Based Deep Learning Method for Machine Vision Inspection



Hao Wu 

Abstract When deep learning methods are applied to the detection of low contrast LCD surfaces, due to the imbalance between positive and negative samples and the difficulty in detecting micro defects with uneven brightness, we propose a method for automatic sample generation and detection based on deep generation network models. Firstly, generate several defect samples by generating adversarial networks to generate an expanded sample dataset. Secondly, the deep generation network is combined with the encoder to form an unsupervised model, and the defective parts of the image are obtained through image comparison. The experimental results confirm that the proposed method can automatically generate LCD image samples, and experiments on Mask R-CNN and unsupervised deep generation network models also confirm the effectiveness of our proposed method.

Keywords Deep learning · Defect detection · Generating adversarial networks

1 Introduction

With the increasing popularity of digital devices such as portable notebooks, mobile phones, and tablets, Liquid Crystal Displays (LCDs) have become a popular choice due to their low power consumption and absence of radiation pollution. This study focuses on the automated visual inspection of microscopic defects such as scratches, inclusions, and pits that cause uneven brightness on low-contrast LCD surfaces, as shown in Fig. 1, the left (a1) (a2) (a3) are defects, and the right (b1) (b2) (b3) are the effects of defect enhancement. These defects are difficult to identify due to their low contrast characteristics, and traditional defect detection methods cannot handle them.

While deep learning methods have shown promise in LCD defect detection [1, 2], the current methods mainly focus on macroscopic defects and are limited by the availability of positive and negative samples for training. In this paper, we propose

H. Wu (✉)

School of Mechanical Engineering, Anhui University of Technology, Maanshan 243032, China
e-mail: hao.wu@ahut.edu.cn

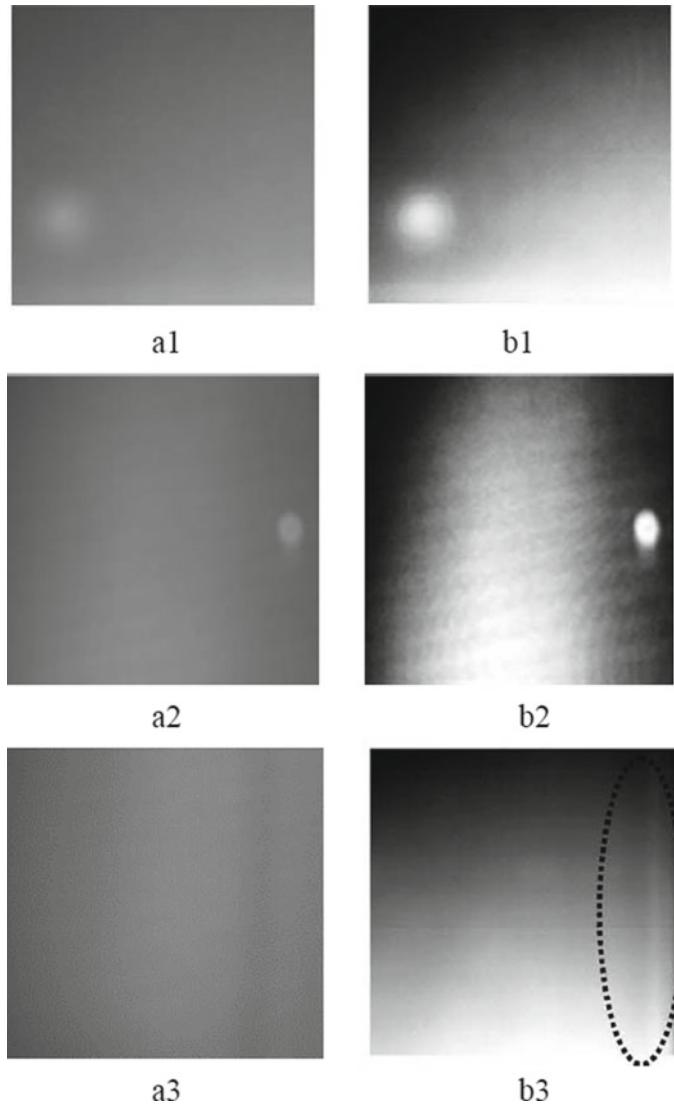


Fig. 1 The uneven brightness defect on the low-contrast LCD surface and its image enhancement effect

using generative adversarial networks to automatically generate LCD image samples for training and testing, thus addressing the issue of sample imbalance.

Methods for detecting surface defects based on machine vision are classified [3] into statistical-based methods, spectrum-based methods [4, 5], subspace-based methods [6, 7], and emerging methods based on deep learning. However, in the LCD manufacturing industry, traditional methods cannot handle low-contrast surface

defect detection with uneven brightness involved in this study. Therefore, more complex machine learning algorithms such as the adaptive threshold method [8], Retinex theory [9], anisotropic diffusion [10], and level set methods [11] were introduced. Deep learning methods such as deep convolutional neural networks [12, 13] have also been increasingly used in LCD defect detection. Remarkable achievements have been made in the tasks of object classification, object detection [14], and object instance segmentation [15, 16].

Current LCD defect detection methods based on deep learning mainly focus on classifying and locating macroscopic defects such as Mura [1, 12, 13]. Some methods, such as the online time series classification method proposed by Yang et al. [13] and the Single Shot multi-box Detector (SSD) network proposed by Ramya et al. [17], have achieved good classification results but low detection accuracy. However, these methods have not yet addressed the detection of LCD micro defects such as tiny scratches, inclusions, and pits.

Furthermore, collecting enough defective samples for training and testing deep learning models is challenging. While it is easy to collect enough qualified samples in the actual LCD manufacturing process, collecting enough defective samples is difficult. To address this issue, current deep learning-based methods mainly use data augmentation to expand the number of defective samples [18, 19]. However, these synthesized defect samples are still different from the actual production samples. Therefore, we propose using generative adversarial networks to automatically generate LCD image samples for training and testing to solve the problem of positive and negative sample imbalance. A deep network model that can automatically generate samples and detection is proposed.

The structure of this paper is as follows. Firstly, the current research status of surface defect detection methods and the problem that the images required for training are difficult to collect are introduced. Then, a method for automatic generation and amplification of samples based on GAN is proposed to solve these problems. Next, an unsupervised model based on the deep generation network is proposed to segment the defect region. Finally, the experimental process and detection results of our proposed method are shown, and the conclusion of the article is given.

2 Automatic Generation Method of Image Samples Based on Generative Adversarial Network

An approach for automatically generating defective samples based on generative adversarial networks (GANs) is proposed to address the problem of unbalanced positive and negative samples required for training deep learning-based surface defect detection methods. It is relatively easy to collect defect-free samples, but it is difficult to collect defective samples. The proposed method only requires a small number of real defective samples to convert non-defective samples to defective samples using the GAN model. The generated defective samples and real samples are combined

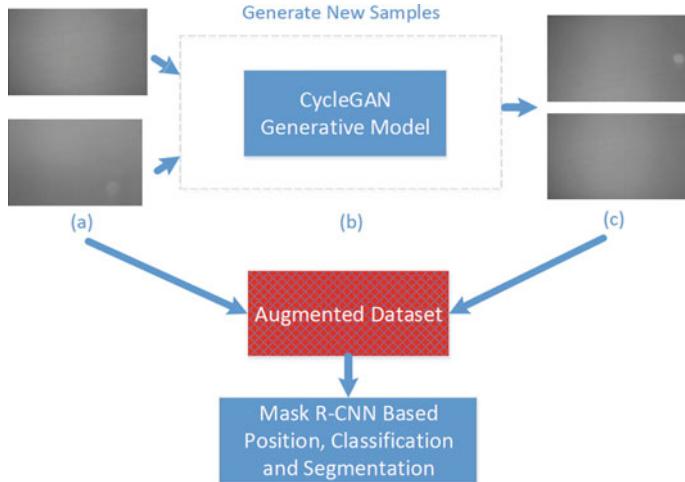


Fig. 2 Image sample generation method based on GAN model

to train the Mask R-CNN model to obtain the best detection results. The method is applied to the experiments of generating LCD defect samples.

The image sample generation method based on the GAN model involves using a randomly collected small set of defect-free images from LCD images as the input set for the CycleGAN model. As shown in Fig. 2, real defect images are used as the target dataset for the model, and CycleGAN learns from the input defect-free images to create a new dataset of defective and non-defective LCD sample images. The newly generated LCD images are combined with existing real LCD images to form an augmented dataset, which is used to train and test the Mask R-CNN model.

2.1 Schematic Diagram of CycleGAN Method

As shown in Fig. 3, the CycleGAN model uses two different style images, X and Y, as input datasets and two generators, G and F. G converts the style of image X to the style of image Y, and F converts the style of image Y to the style of image X. CycleGAN converts the style of image X to the style of image Y through G and can still be converted back to the style of X after passing through F, ensuring content consistency. Similarly, CycleGAN converts the style of image Y to the style of image X through F and can still convert back to the style of Y after passing through G, ensuring content consistency.

Cyclegan loss function:

$$\text{Loss} = \text{Loss}_{GAN} + \text{Loss}_{cycle} \quad (1)$$

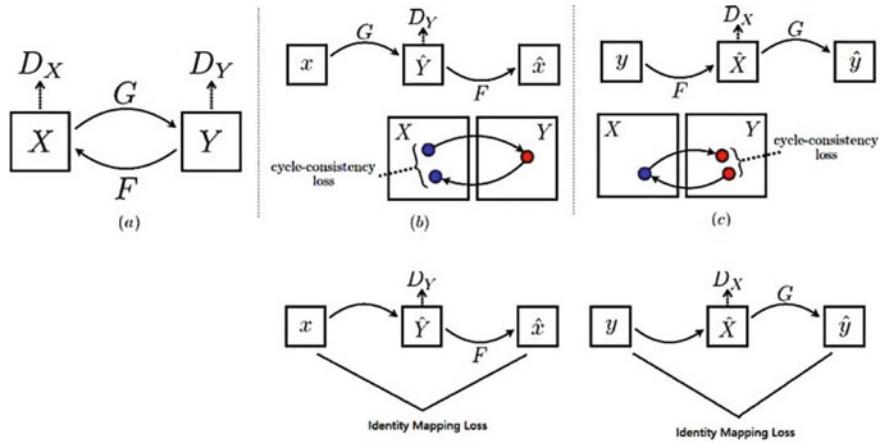


Fig. 3 Cyclegan model diagram

$Loss_{GAN}$ ensures that the generator and the discriminator evolve each other, thereby ensuring that the generator can generate more realistic pictures.

$Loss_{GAN}$ ensures that the output image of the generator is different from the input image, but the content is the same. Specific:

$$\begin{aligned} Loss_{GAN} &= L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, X, Y) \\ &= E_{y \sim p_{data}(y)} [\log D_Y(y)] + E_{x \sim p_{data}(x)} [\log(1 - D_Y(G(x)))] \\ &\quad + E_{x \sim p_{data}(x)} [\log D_X(x)] + E_{y \sim p_{data}(y)} [\log(1 - D_X(F(y)))] \end{aligned} \quad (2)$$

$$Loss_{cycle} = E_{x \sim p_{data}(x)} [\|F(G(x)) - x\|_1] + E_{y \sim p_{data}(y)} [\|G(F(y)) - y\|_1] \quad (3)$$

$$MinimizeLoss = MinimizeLoss_{GAN} + MinimizeLoss_{cycle} \quad (4)$$

When training the generator, the D_x and D_y parameters are fixed, and only the parameters of G and F are adjustable, so:

$$\begin{aligned} MinimizeLoss_{GAN} &= Minimize E_{x \sim p_{data}(x)} [\log(1 - D_Y(G(x)))] \\ &\quad + Minimize E_{y \sim p_{data}(y)} [\log(1 - D_X(F(y)))] \\ &= Maximize E_{x \sim p_{data}(x)} [\log D_Y(G(x))] \\ &\quad + Maximize E_{y \sim p_{data}(y)} [\log D_X(F(y))] \end{aligned} \quad (5)$$

which is:

Adjust the parameters of G so that the higher the score $D_y(G(x))$ of D_y on the picture $G(x)$ generated by G , the better. Adjust the parameters of F so that the higher the score $D_x(F(y))$ of D_x on the picture $F(y)$ generated by F , the better.

MinimizeLoss_{GAN} ensures that the pictures generated by the generator are more and more realistic (the style is more and more like another type of pictures).

The purpose of *MinimizeLoss_{cycle}* is to make $F(G(x)) = x$ and $G(F(y)) = y$. Taking $F(G(x)) = x$ as an example, the premise of satisfying this purpose is that $G(x) \approx x$, otherwise it is impossible for the generator F to restore the $G(x)$ that has nothing to do with x to x . Therefore, *MinimizeLoss_{cycle}* ensures that the content of the pictures generated by the generator is roughly unchanged.

When training the discriminator, the G and F parameters are fixed, and only the parameters of D_x and D_y are adjustable. At this time, we are not *MinimizeLoss* but *MaximizeLoss*.

$$\begin{aligned}
\text{MaximizeLoss} &= \text{MaximizeLoss}_{GAN} \\
&= \text{Maximize}_{GAN}(F, D_X, X, Y) \\
&= \text{Maximize}_{x \sim p_{data(x)}} [\log D_X(x)] \\
&\quad + \text{Maximize}_{y \sim p_{data(y)}} [\log(1 - D_X(F(y)))] \\
&= \text{Maximize}_{x \sim p_{data(x)}} [\log D_X(x)] \\
&\quad + \text{Minimize}_{y \sim p_{data(y)}} [\log D_X(F(y))] \tag{6}
\end{aligned}$$

That is, when training the discriminator D_x , the value of $D_x(x)$ should be maximized, and at the same time, the value of $D_x(F(y))$ should be minimized (let the discriminator give the generated image $F(y)$ low score), so that Improve the discrimination ability of the discriminator.

3 Automatic Defect Image Inspection Based on Generative Adversarial Networks

In this study, we also propose a novel approach for automatically acquiring image defect areas to train deep learning models for defect detection. The proposed method is illustrated in Fig. 4. Initially, an unsupervised Generative Adversarial Network (GAN) model is trained on a set of normal images. The GAN learns to generate synthetic images that closely resemble the real images. The model then calculates an anomaly score for a given test image to determine whether it contains any defects. The image defect can be identified by comparing the synthetic image generated by the GAN with the input image.

3.1 Unsupervised Generative Adversarial Network Model

The GAN model used in this study is unsupervised, meaning that only a large set of normal images is required for training, without any image annotation. First, the

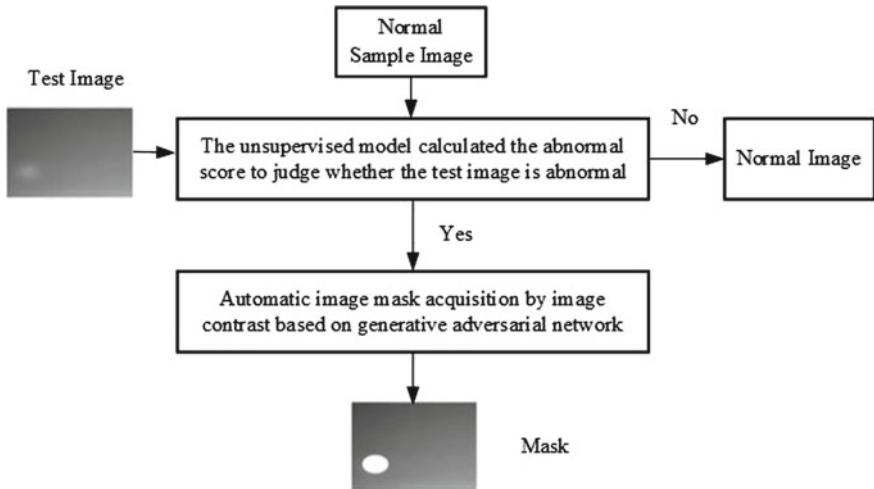


Fig. 4 Automatic image mask acquisition method

normal (i.e., defect-free) images are selected for training. The unsupervised learning method captures the inherent variability in the image data used in training. The encoder is trained to quickly map an image to its corresponding hidden space representation, enabling efficient evaluation of whether a new test image belongs to the normal image category.

Figure 5 shows the training and testing process of the unsupervised GAN model. During training, the generator generates synthetic images based on random noise in the hidden space. The discriminator adjusts the generator's parameters to make the distribution of the synthetic images as close as possible to that of the real images. The encoder model is then trained, and the trained model is saved. During testing, the model calculates an anomaly score for a test image and generates synthetic images that closely resemble the input image to identify possible defects.

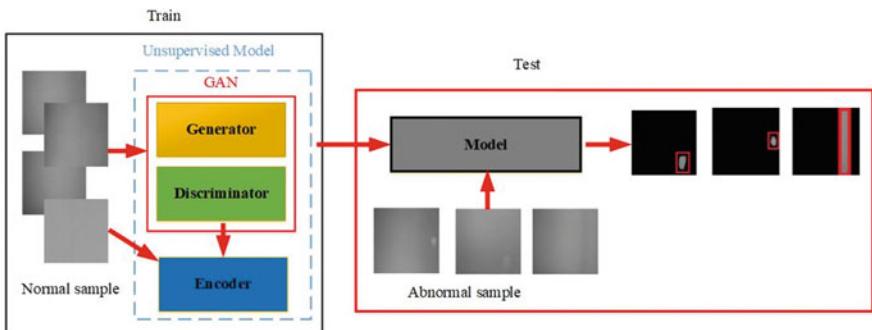


Fig. 5 Training and testing of unsupervised GAN model

The architecture of the training encoder network is shown in Fig. 6. The encoder network has a fixed weight and is trained alongside the generator and discriminator. The encoder learns the mapping of the hidden space representation and maps it back to the image space using the generator. When training the encoder, the mapping of the hidden space Z can be obtained, and the hidden space Z is then mapped back to the image space $G(E(x))$ through the generator. The objective is to minimize the mean square error between the input image x and the reconstructed image.

$$L_{izi}(x) = \frac{1}{n} \|x - G(E(x))\|^2 \quad (7)$$

where n is the number of pixels of the input image x and $G(E(x))$ is the image generated by the generator and encoder of the input image x .

The anomaly score of the test image can be calculated based on the error between the test image and the generated image:

$$\text{Anomaly score} = \|x - G(E(x))\| \quad (8)$$

3.1.1 Determination of Threshold T

The anomaly score of a test image is calculated based on the error between the test image and the synthetic image. The test image is deemed defective if the anomaly score exceeds a threshold value. The threshold is determined by analyzing the abnormal scores of normal and defective test images and finding the overlap between their probability density histograms.

$$T = f_1(x), \text{if } f_1(x) = f_2(x) \quad (9)$$

where T is the threshold, $f_1(x)$ is the abnormal fraction distribution of normal images, and $f_2(x)$ is the abnormal fraction distribution of abnormal images.

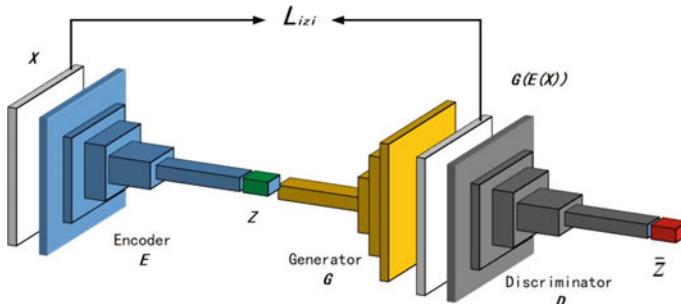


Fig. 6 Encoder network structure

3.1.2 Difference Between Generated and Test Images

The difference between the synthetic and test images can be used to identify possible defects. The residual image is obtained by subtracting the input image from the synthetic image, which yields a mask corresponding to the input image.

$$R = \max(\|x_r - G(E(x_r))\|, \|x_g - G(E(x_g))\|, \|x_b - G(E(x_b))\|) \quad (10)$$

where R is the residual image; x_r , x_g , and x_b are the red, green, and blue colors of input images, respectively; and $G(E(x_r))$, $G(E(x_g))$, and $G(E(x_b))$ are the images generated by the encoder and generator. The result of the image comparison is the mask corresponding to the input image.

4 Experimental Results and Analysis

The proposed GAN-based defect generation method was evaluated on both labeled LCD defects and defect-free LCD images. The hardware and software configuration information includes GPU Nvidia RTX4000 and Python 3.

4.1 Experiments and Results Foe CycleGan Based Dataset Augmentation

To meet the training requirements for the original dataset, data augmentation techniques such as rotating and mirroring the images were used. However, the resulting dataset was still limited, and the GAN-based data sample generation method was employed to expand the dataset. As there were a small number of existing LCD sample images with defects, CycleGAN was used to expand the sample dataset. The GAN adversarial generation network generated new data sets based on the characteristics of the small number of data sets to achieve the purpose of dataset expansion when the dataset was small and insufficient.

The data set samples with defective LCD images and normal LCD images were put into training and testing, respectively. After training, the fake samples produced by the test were combined with the original samples, and then a new round of training and testing was carried out to obtain a large number of defect samples of different styles.

In this study, the learning rate was set to 0.0002, and the regularization λ controlled the importance between the adversarial loss and the cycle-consistency loss. A small value was set for λ to ensure that the synthetic defects were as close to the real

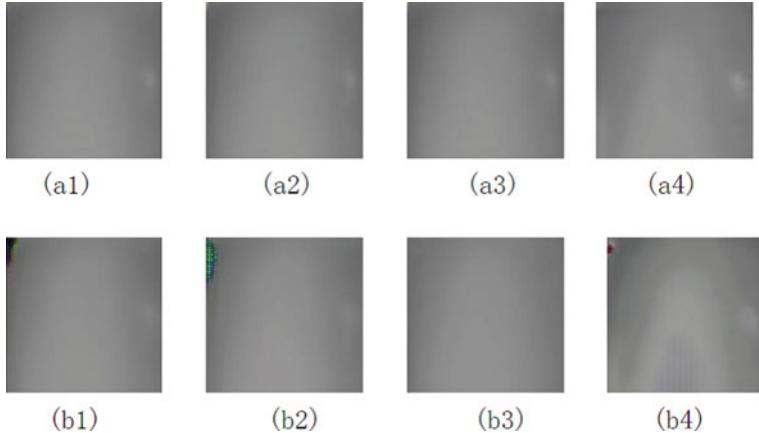


Fig. 7 Comparison of different λ values of CycleGAN. The defect images generated by (a1)–(a4) correspond to $\lambda = 1, 10, 45, 100$, respectively. The defect-free images generated by (b1)–(b4) correspond to $\lambda = 1, 10, 45, 100$, respectively

defects as possible. The choice of model parameters considered not only the background pattern reconstruction but also the generation of effective defects with sufficient contrast. A large-scale parameter test was performed, and a suitable parameter λ was found to achieve a balance between background reconstruction and defect reconstruction. $\lambda = 45$ was adopted in this study to balance the overall performance between defect synthesis and background preservation.

The results in Fig. 7 show that too large λ value cannot effectively generate defects, and too small λ value has poor effect of reconstructing background texture. Combined with practical requirements to balance the overall performance between defect synthesis and background preservation, $\lambda = 45$ was adopted in this study.

Here, various types of LCD samples are generated using the CycleGAN model. Figure 8a–b are the real defect LCD samples, and Fig. 8c is a real defect-free LCD sample used as input to train the CycleGAN model.

Figure 9 is the generated LCD image, Fig. 9a–b is the generated defective sample, and Fig. 9c is the generated non-defective sample. It can be clearly seen that CycleGAN can swap the styles of LCD images despite very different samples and different backgrounds.

4.1.1 Use the Mask RCNN Model for Recognition

For recognition, the generated defect sample images were marked and classified using the Mask R-CNN model. The dataset was prepared using the generated defect sample images. The labelme tool was used to label and segment the image, mark the background and target, and automatically cut the target area image. The cut

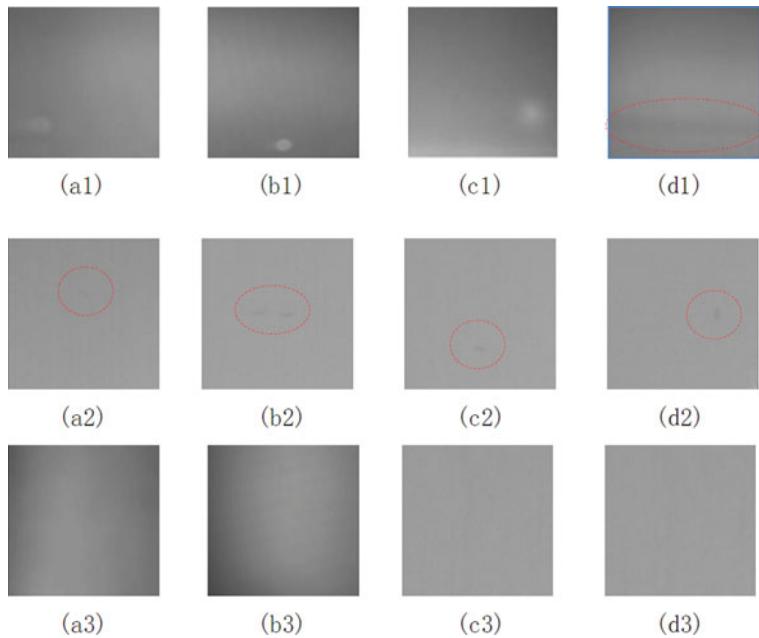


Fig. 8 Original real LCD image sample

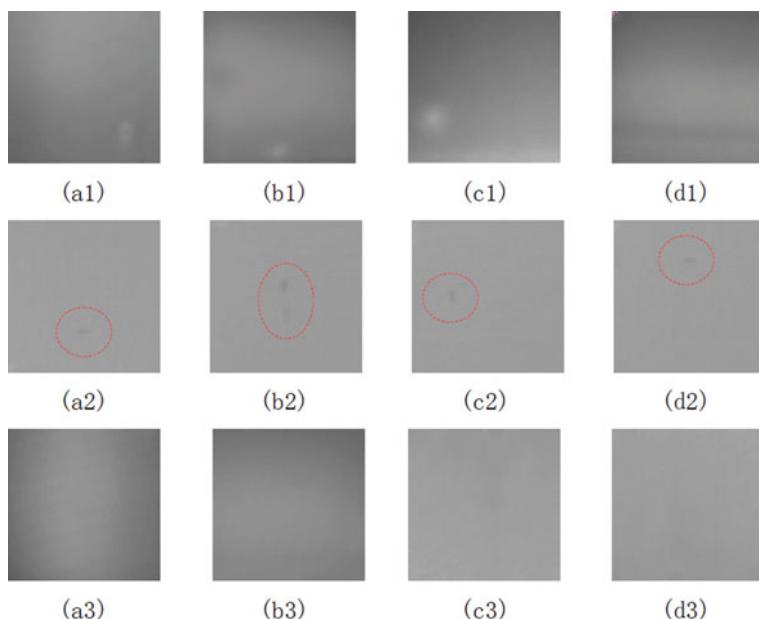


Fig. 9 Generated LCD image sample

Table 1 Recognition rate distribution in each group of test results

Recognition rate	<0.8	<0.9	<0.95	>0.95
Group 1 (50)	2	4	2	12
Group 2 (100)	0	2	6	12

image was masked and converted into a mask. The Mask R-CNN model was used for training and testing the trained model.

The detailed test results are shown in Table 1 below. It can be seen from the Table 1 that most of the test results in the first group have a recognition rate above 0.95, but there are also 6 of them that are less than 0.9. The test results showed that the average recognition rate of the first group is 0.921, and the average recognition rate of the second group is 0.959.

Mask R-CNN is used to train and test the generated LCD data set. From the expand of the data set, the modification of training parameters to the test, all are indispensable. The compares results of training the model with 50 samples and 100 samples shows: the average recognition rate of the test after training with 100 samples is higher than the average recognition rate of the test training with 50 samples, and the test results of the model trained with 100 samples do not appear that the recognition rate is too low, the more training samples, the better the test recognition effect will be.

4.2 Experimental Results and Analysis for Automatic Defect Image Inspection Based on Generative Adversarial Networks

In this study, we used the MVTec AD dataset [18] for training and testing an automatic defect image inspection system based on Generative Adversarial Networks (GANs). The dataset contains over 5,000 high-resolution images divided into 15 object and texture categories, each comprising a training set of defect-free images and a test set containing images with various defects and without defects. The dataset's representative samples (wood, bottle, and leather,) are shown in Table 2.

Table 2 Several types of representative samples in datasets

Category	Number of training samples		Number of testing samples	
	Normal	Defect	Normal	Defect
Wood	246	0	18	55
Bottle	208	0	19	60
Leather	244	0	31	87

4.2.1 Probability Distribution Statistics of Anomaly Scores

Using the aforementioned abnormal score calculation method, density histograms were drawn of the abnormal scores for the normal and defect images of each type of test sample. Samples for several categories (wood, bottle, and leather) are shown in Fig. 10. As shown in Fig. 10, the normal images and defect images have obvious aggregation. Thus, the overlapping position of the two is regarded as their threshold value.

4.2.2 Performance Evaluation of Proposed Method

Next, we evaluated the performance of the algorithm in classifying the results for each image. An image to be tested can be classified into two categories (defective or non-defective), and a prediction can produce four different results. True positive (TP) and true negative (TN) indicate correct classification results. A false positive (FP) means that a sample is actually unqualified, but the predicted result is qualified. Likewise, a false negative (FN) means that a sample is actually qualified, but the predicted result is unqualified. Based on these four values, the TP rate (TPR), FP rate (FPR), and precision (PRC) can be calculated. A well-known visualization of a classifier's accuracy is the receiver operator characteristic (ROC) curve, which plots the two metrics FPR and TPR against each other. The area under the ROC curve (AUC) can then be calculated, as shown in Fig. 11. Our experimental results show that the method proposed in this paper can accurately segment the image target object.

$$TPR = \frac{TP}{TP + FN} \quad (11)$$

$$FPR = \frac{FP}{FP + TN} \quad (12)$$

$$PRC = \frac{TP}{TP + FP} \quad (13)$$

5 Conclusion

When the deep learning method is used to detect microscopic defects with uneven brightness on the low-contrast LCD surface, we propose a GAN based method for automatic sample generation to address the imbalance of positive and negative samples. Then we also proposed an unsupervised GAN model to automatically detect the image defect area.

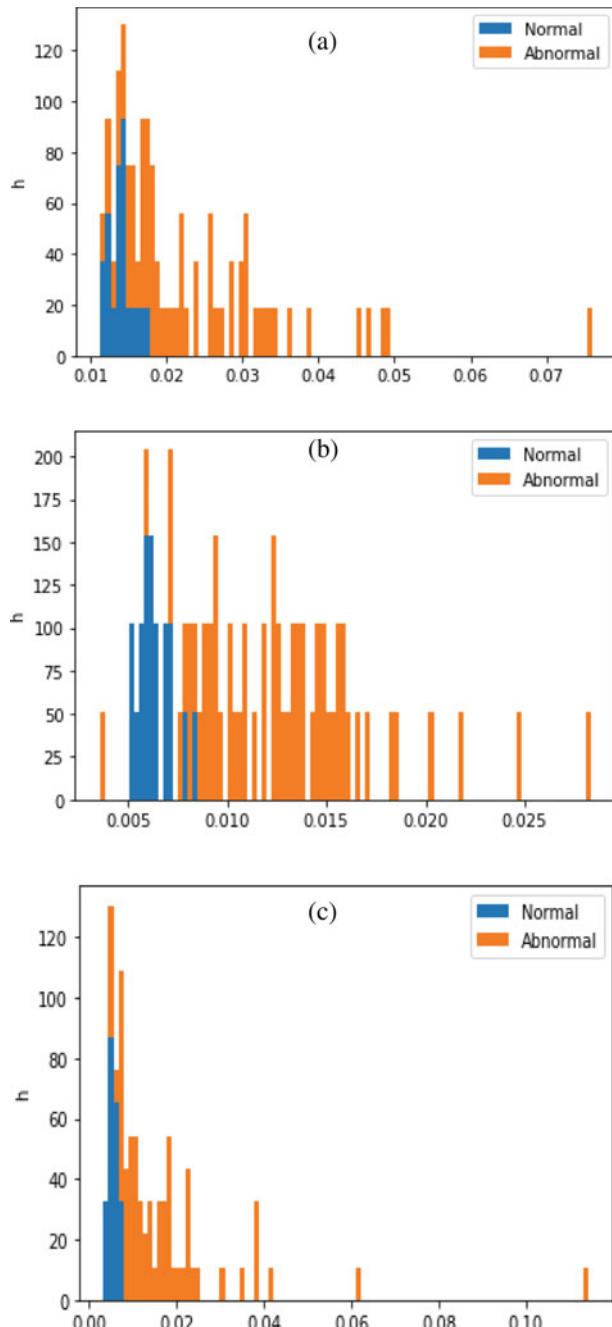
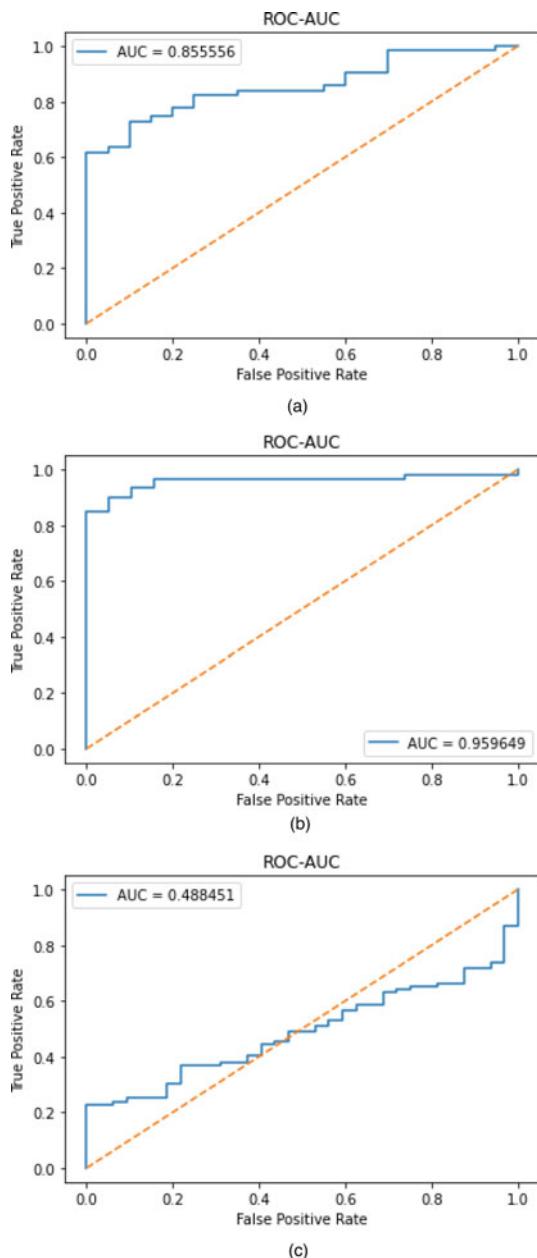


Fig. 10 Density histogram of abnormal scores for normal and defect images. **a** Bottle. **b** Wood. **c** Leather

Fig. 11 ROC curves for different types of images.
a Bottle. **b** Wood. **c** Leather



Through experiments, the effectiveness of our proposed method is verified, not only to obtain enough samples for training deep learning methods, but also the experimental results on the detection of microscopic defects on the low-contrast LCD surface also confirm the obtained image samples and image defect masks with good detection accuracy, the proposed method can also be used to fields requiring image sample augmentation and image annotation.

Acknowledgements This research was supported in part by the China National Key Research and Development project (2017YFE0113200), Anhui Provincial Natural Science Foundation (2108085ME166), Natural Science Research Project of Universities in Anhui Province (KJ2021A0408), and Open Project of Anhui Province Key Laboratory of Special and Heavy Load Robot (TZJQR007-2021).

References

1. Lu, H. P., & Su, C. T. (2021). CNNs combined with a conditional GAN for Mura defect classification in TFT-LCDs[J]. *IEEE Transactions on Semiconductor Manufacturing*, 34(1), 25–33.
2. Kim, M., Lee, M., & An, M. (2020). Effective automatic defect classification process based on CNN with stacking ensemble model for TFT-LCD panel[J]. *Journal of Intelligent Manufacturing*, 31(5), 1165–1174.
3. Xie, X. (2008). A review of recent advances in surface defect detection using texture analysis techniques[J]. *ELCVIA: Electronic Letters on Computer Vision and Image Analysis*, 7(3), 1–22.
4. Li, Y., Luo, H., & Yu, M. (2019). Fabric defect detection algorithm using RDPSO-based optimal Gabor filter[J]. *The Journal of the Textile Institute*, 110(4), 487–495.
5. Zhou, X., Wang, Y., & Zhu, Q. (2019). A surface defect detection framework for glass bottle bottom using visual attention model and wavelet transform[J]. *IEEE Transactions on Industrial Informatics*, 16(4), 2189–2201.
6. Yufeng, S., Dali, Z., & Junhua, Z. (2020). Analysis of textile defects based on PCA-NLM[J]. *Journal of Intelligent and Fuzzy Systems*, 38(2), 1463–1470.
7. Ahmad, J., Akula, A., & Mulaveesala, R. (2019). An independent component analysis based approach for frequency modulated thermal wave imaging for subsurface defect detection in steel sample[J]. *Infrared Physics and Technology*, 98, 45–54.
8. Kim, S. Y., Song, Y. C., Jung, C. D., & Park, K. H. (2011). Effective defect detection in thin film transistor liquid crystal display images using adaptive multi-level defect detection and probability density function[J]. *Optical Review*, 18(2), 191–196.
9. Zhang, Y., Zhang, Y., & Gong, J. (2020). A LCD screen Mura defect detection method based on machine vision[C]. In *2020 Chinese control and decision conference (CCDC)* (pp. 4618–4623). IEEE.
10. Chen, C. S., Weng, C. M., & Tseng, C. C. (2018). An efficient detection algorithm based on anisotropic diffusion for low-contrast defect[J]. *The International Journal of Advanced Manufacturing Technology*, 94(12), 4427–4449.
11. Yang, H., Song, K., & Mei, S. (2018). An accurate mura defect vision inspection method using outlier-prejudging-based image background construction and region-gradient-based level set[J]. *IEEE Transactions on Automation Science and Engineering*, 15(4), 1704–1721.
12. Mei, S., Yang, H., & Yin, Z. (2017). Unsupervised-learning-based feature-level fusion method for Mura defect recognition[J]. *IEEE Transactions on Semiconductor Manufacturing*, 30(1), 105–113.

13. Yang, H., Mei, S., Song, K., Tao, B., & Yin, Z. (2018). Transfer-learning-based online Mura defect classification[J]. *IEEE Transactions on Semiconductor Manufacturing*, 31(1), 116–123.
14. Ren, S., He, K., & Girshick, R. (2016). Faster r-cnn: Towards real-time object detection with region proposal networks[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149.
15. Huang, Z., Huang, L., & Gong, Y., et al. (2019). Mask scoring r-cnn[C]. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6409–6418).
16. Bolya, D., Zhou, C., & Xiao, F. (2019). Yolact: Real-time instance segmentation[C]. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 9157–9166).
17. Singh, R., Kumar, G., & Sultania, G. (2019). Deep learning based Mura defect detection[J]. *EAI Endorsed Transactions on Cloud Systems*, 5(15), 1–7.
18. Radford, A., Metz, L., Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks[J]. arXiv preprint [arXiv:1511.06434](https://arxiv.org/abs/1511.06434)
19. Chen, X., Girshick, R., & He, K. (2019) Tensormask: A foundation for dense object segmentation[C]. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 2061–2069).

Generative Adversarial Networks for Stain Normalisation in Histopathology



Jack Breen, Kieran Zucker, Katie Allen, Nishant Ravikumar,
and Nicolas M. Orsi

Abstract The rapid growth of digital pathology in recent years has provided an ideal opportunity for the development of artificial intelligence-based tools to improve the accuracy and efficiency of clinical diagnoses. One of the significant roadblocks to current research is the high level of visual variability across digital pathology images, causing models to generalise poorly to unseen data. Stain normalisation aims to standardise the visual profile of digital pathology images without changing the structural content of the images. In this chapter, we explore different techniques which have been used for stain normalisation in digital pathology, with a focus on approaches which utilise generative adversarial networks (GANs). Typically, GAN-based methods outperform non-generative approaches but at the cost of much greater computational requirements. However, it is not clear which method is best for stain normalisation in general, with different GAN and non-GAN approaches outperforming each other in different scenarios and according to different performance metrics. This is an ongoing field of study as researchers aim to identify a method which efficiently and effectively normalises pathology images to make AI models more robust and generalisable.

Keywords Computational pathology · Computer-aided diagnosis · Generative models · Robustness · Style transfer

J. Breen (✉) · N. Ravikumar

Centre for Computational Imaging and Simulation Technologies in Biomedicine (CISTIB),
School of Computing, University of Leeds, Leeds, UK
e-mail: scjb@leeds.ac.uk

K. Zucker

Leeds Cancer Centre, St. James's University Hospital, Leeds, UK

K. Allen · N. M. Orsi

School of Medicine, Leeds Institute of Medical Research at St. James's, University of Leeds,
Leeds, UK

1 Histopathology

Histopathology is the microscopic evaluation of tissue for medical diagnosis. It is an essential part of the diagnostic pathway for many diseases, including autoimmune disorders, infections, and cancers. Tissue samples are taken either as biopsies or larger tissue resections, which are then fixed in formalin, embedded in paraffin, sectioned, and stained to create pathology slides. This process prevents tissue degradation and allows for long-term storage. Samples are occasionally flash-frozen, a faster process which allows pathologists to provide rapid information during surgery, but at the expense of increased cell damage and inferior staining quality. Samples are most commonly stained with haematoxylin and eosin (H&E), which make cell nuclei appear blue and cytoplasm pink, with other cellular structures appearing varying shades of purple, pink, red, and blue. Immunohistochemistry (IHC) is another common staining technique which is used to identify specific antigens (proteins) in a tissue sample, which can aid in distinguishing between differential diagnoses and making prognostic predictions.

Historically, pathologists analysed tissue samples using light microscopy; however, this is increasingly being replaced by a digital workflow where tissue slides are scanned at high resolution to generate whole slide images (WSIs) which can be visually assessed on a computer screen. Digitisation can drastically improve the efficiency of the diagnostic process [5, 22] with minimal impact on diagnostic decisions [42], though the high start-up costs and technical requirements have slowed the rate of adoption. While the digital pathology workflow has primarily been developed for logistical and long-term financial reasons, it has also revolutionised diagnostic AI by creating large digital repositories of histopathology images. Models have been developed for a wide array of diagnostic and prognostic tasks [1, 10], with AI researchers aiming to improve the accuracy and efficiency of the interpretation of pathology specimens. Diagnostic accuracy is essential to ensure that patients receive optimal treatment. Diagnostic efficiency is equally vital as there is a global shortage of pathologists, with some countries having access to only a small fraction of the number of pathologists available to others [60], causing a diagnostic bottleneck. Diagnostic delays can have catastrophic consequences, with a four-week delay in cancer treatment being associated with a 10% reduction in survival [21]. Demand for pathologists is expected to continue increasing due to the ongoing global population growth and ageing trends.

Current AI models have limited clinical utility, with the United States Food and Drug Administration (FDA) having only approved one *AI-enabled medical device* in digital pathology imaging. This tool classifies whether prostate biopsies contain malignant cells and indicates the most likely affected area [15]. While this is a success story for digital pathology AI, the task of prostate biopsy malignancy classification has many enabling traits: it is a very common disease [53], biopsy slides contain orders of magnitude less tissue than resection slides, there are only two possible classes, and it is a relatively straightforward task for human pathologists, who achieve over 90% accuracy [15]. The high incidence rate of prostate cancer makes it possible

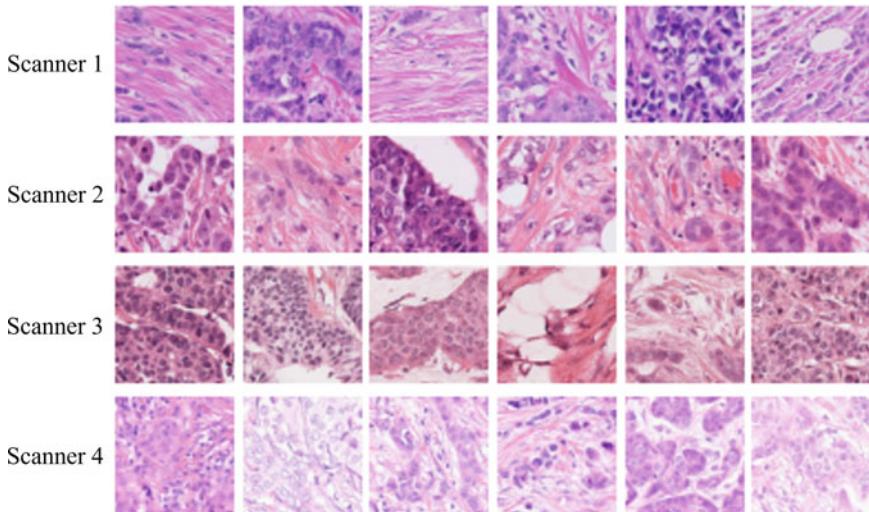


Fig. 1 Examples of visual variation caused by different scanners from the MIDOG 2021 Challenge training set [4], where each tissue sample was processed in the same laboratory following the same protocol, and then digitised with one of four available scanners. Image adapted from [11]

to collect vast quantities of varied data, and the relatively small size of biopsy samples makes it possible to train a model with a huge number of samples, allowing for the development of a robust model. In scenarios where it is not possible to train a model with such varied data, model robustness is a critical barrier to clinical implementation.

Digital pathology slides vary visually due to sample processing differences [36] (e.g. cut-up, fixation, and staining protocol), biological differences [3] (e.g. tissue type, genetics), and digitisation differences [45] (e.g. scanner, magnification, file formatting). Examples of the variation caused by different scanners are shown in Fig. 1. Many factors will not vary significantly over short periods in a single pathology lab, meaning that single-centre data will typically be more homogeneous than multi-centre data, and hence that models trained on single-centre data will be less robust to variability. Such models are likely to generalise poorly to data from different data centres, and even to data from a single centre over time due to changes in tissue processing and digitisation. Stain normalisation, a form of style transfer, can improve model generalisability by reducing the variability of digital pathology data.

2 Style Transfer

Style transfer methods aim to adjust the visual style of an image to match that of another image (or set of images) while retaining the original structure. Styles may be artificial, such as the painting style of a specific artist, or realistic, such as different

seasonal colours in landscape photography. A toy example is converting “horses to zebras” due to these animals sharing similar physiques but having distinct visual styles [63]. We may refer to these images as coming from different *domains*, which in histopathology will often refer to different data centres which have distinct tissue processing and digitisation protocols.

Style transfer may be supervised or unsupervised. Supervised training requires paired images showing the same content in different domains for training. Unsupervised training does not use paired images, so is often used in scenarios where it is impossible or impractical to collect perfectly paired images, such as the horse-to-zebra model and art-to-photograph models presented in the original CycleGAN paper [63]. It is not common to have paired histopathology data from different domains. Many domain differences are impossible to capture in paired histopathology data, such as the tissue staining procedure, which cannot be repeated for the same tissue sample. The variation of different scanners can be captured by repeatedly scanning the same sample, though this is rarely done as it increases costs without a direct benefit to patient care. Approximate paired data can be produced by differently processing consecutive tissue samples, though this is also rare. Supervised techniques can still be used when paired data is unavailable through the creative generation of artificial paired images, for example by taking the greyscale and full-colour versions of the same histopathology slide as a pair. This can be beneficial as supervised methods tend to be more computationally efficient and require less data than unsupervised methods.

2.1 Generative Models

Many style transfer models are based on generative adversarial networks (GANs) [20]. These are two-part models containing a generator, which attempts to create new samples from a given distribution, and a discriminator, which attempts to distinguish real samples from generated ones. The standard GAN loss function may be expressed as:

$$\mathcal{L}_{GAN}(G, D_G) = \mathbb{E}_{\mathbf{x} \sim p_{data}} [\log(D_G(\mathbf{x}))] + \mathbb{E}_{\mathbf{x} \sim p_g} [\log(1 - D_G(\mathbf{x}))], \quad (1)$$

for generator G , corresponding discriminator D_G , real data distribution p_{data} , and generated data distribution p_g . This *adversarial loss* represents the separation between real images and generated images, with the generator aiming to minimise the loss against its adversary, the discriminator, which aims to maximise it.

GANs are not the only generative models used in style transfer. Variational autoencoders (VAEs) [29] are encoder-decoder networks which impose a prior distribution onto the latent space. A latent loss is used during training to make the posterior latent distribution similar to the prior distribution. The standard VAE aims to exactly reproduce an input image, but it can be adjusted to produce style-transferred images, typically by combining it with a GAN to create a VAE-GAN, where the decoder of a VAE is used as the generator of a GAN, as shown in Fig. 2.

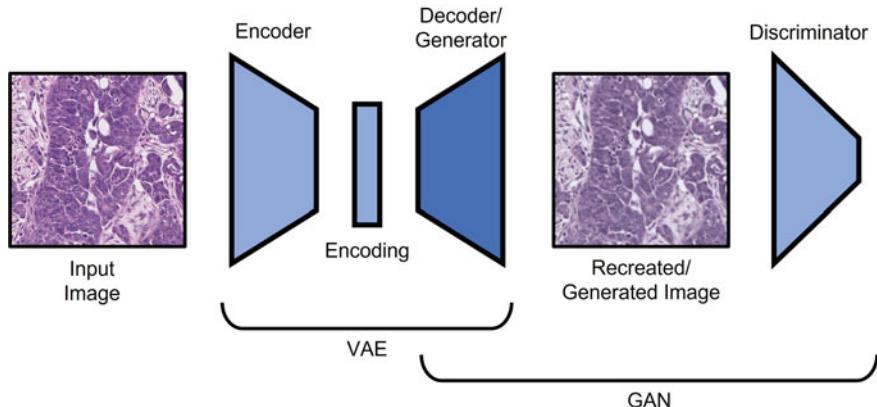


Fig. 2 Simplified VAE-GAN architecture, with the VAE decoder used as the GAN generator. Figure adapted from [30]

Like VAEs, Gaussian Mixture Models (GMMs) assume an underlying distribution within the data, specifically a combination of multiple Gaussian distributions. In the context of style transfer, the typical approach is to iteratively adjust the colours of the input image using expectation maximisation to better fit the prior distribution. This approach has been applied to stain normalisation for many years [38], with newer approaches combining GMMs with deep learning [61].

2.2 Similarity Metrics

Generative style transfer methods can be evaluated in many ways. Some researchers focus on evaluating the effects of normalisation on a downstream AI task, while others try to directly evaluate the quality of synthetic images or the similarity between generated and real images. Stain normalisation can also be assessed qualitatively, for example by pathologists evaluating the quality of images and the severity of artifacts [40].

Different quantitative similarity metrics are suitable in different scenarios, for example, if a model attempts to create an exact reconstruction of an input image then it may be appropriate to compare the original to the reconstruction using a holistic similarity measure such as Euclidean distance, L1 distance, or mean squared error. However, these distance metrics are less appropriate when comparing a stylised image to an original image, as the stylised image should not be an exact reconstruction of the input image. It may be more appropriate to use specific metrics such as the structural similarity index measure (SSIM), which focuses on shape but not colour. Maximising SSIM helps to find a model which retains the structural elements of an image, though

this metric cannot quantify the change in style, which can be assessed qualitatively or through colour distribution comparisons (e.g. Kullback-Leibler divergence).

Instead of directly comparing images, some metrics are based on the features generated by a convolutional neural network encoder. This includes the Fréchet inception distance (FID), which compares the distribution of encoded features from real and generated images, and the learned perceptual image patch similarity (LPIPS), which is explicitly trained to mimic human judgements. These metrics require careful training and are more computationally complex than non-trainable metrics, but they remain popular as they can offer more thorough assessments of similarity.

2.3 Loss Functions

Generative style transfer methods are often complex systems containing many different functions, each of which needs to be trained differently, leading to the use of multiple loss functions during the training of an individual network. These loss functions are diverse, although they typically include an adversarial loss and a reconstruction loss. The adversarial loss (such as the standard GAN loss in Eq. 1) trains the model to produce realistic samples which fit within the distribution of the real images, and the reconstruction loss trains the model to retain certain information from the input image in the generated image. These are both important, as without the adversarial loss the model may not generate realistic images, and without the reconstruction loss the model may generate images which do not represent the input image. Models may also include a feature-preserving loss, which works the same as a reconstruction loss but is applied in an encoded feature space, and a conditional contrastive loss, which aims to cluster samples into groups with shared class labels in feature space.

Reconstruction Losses

Reconstruction losses are applied to retain information between an input image and a generated image. They are often based on the similarity metrics discussed in Section 2.2, and can be as simple as taking a Euclidean distance between the input image \mathbf{x} and generated image $G(\mathbf{x})$ [13]:

$$\mathcal{L}_r(G) = \mathbb{E}_{\mathbf{x} \sim p_{data(x)}}[||\mathbf{x} - G(\mathbf{x})||_2]. \quad (2)$$

Such a basic reconstruction loss can be prone to generating blurry areas, with the L1 loss mitigating this slightly but not solving the issue [24]. These simple losses disincentivise any change between the original and generated image, so some researchers use more targeted metrics, such as SSIM, in their losses [34]:

$$\mathcal{L}_r(G) = \mathbb{E}_{\mathbf{x} \sim p_{data(x)}}[1 - SSIM(\mathbf{x}, G(\mathbf{x}))]. \quad (3)$$

Feature-Preserving Losses

Feature-preserving losses work the same as reconstruction losses but are applied in an encoded feature space, rather than being applied directly to images. A simple Euclidean distance could still be applied as

$$\mathcal{L}_{fp}(G) = \mathbb{E}_{\mathbf{x} \sim p_{data(x)}} [||\mathcal{F}(\mathbf{x}) - \mathcal{F}(G(\mathbf{x}))||_2], \quad (4)$$

where $\mathcal{F}(\mathbf{x})$ represents the features extracted from image \mathbf{x} . An example which has been used in this field is [13]:

$$\begin{aligned} \mathcal{L}_{fp}(G) &= \mathbb{KL}[\mathcal{F}(\mathbf{x}) || \mathcal{F}(G(\mathbf{x}))] \\ &= \mathbb{E}_{\mathbf{x} \sim p_{data(x)}} [\log(\mathcal{F}(\mathbf{x})) - \log(\mathcal{F}(G(\mathbf{x})))], \end{aligned} \quad (5)$$

where \mathbb{KL} is the Kullback-Leibler divergence. A similar loss is typically used in VAEs to impose the prior distribution onto the feature space [29].

Conditional Losses

Conditional GANs (cGANs) [41] leverage extra information, with the model depending on certain class labels associated with the images. This can be implemented in different ways, with the original cGAN approach [41] adjusting the standard GAN loss to give:

$$\mathcal{L}_{cGAN}(G, D_G) = \mathbb{E}_{\mathbf{x} \sim p_{data}} [\log(D_G(\mathbf{x}|\mathbf{y}))] + \mathbb{E}_{\mathbf{x} \sim p_g} [\log(1 - D_G(\mathbf{x}|\mathbf{y}))], \quad (6)$$

where \mathbf{y} are the class labels. Conditional losses may instead be applied in the feature space, where a conditional contrastive loss aims to create a feature space in which samples are tightly clustered by their input label according to a chosen distance/similarity metric, such as the Euclidean distance or cosine similarity. When using cosine similarity, the loss can be formulated as [27]:

$$\mathcal{L}_{ccl}(\mathbf{x}_i, y_i; t) = -\log \left(\frac{\exp(l(\mathbf{x}_i)^\top e(y_i)/t) + \sum_{k=1}^m \mathbb{1}_{y_k=y_i} \cdot \exp(l(\mathbf{x}_i)^\top l(\mathbf{x}_k)/t)}{\exp(l(\mathbf{x}_i)^\top e(y_i)/t) + \sum_{k=1}^m \mathbb{1}_{k \neq i} \cdot \exp(l(\mathbf{x}_i)^\top l(\mathbf{x}_k)/t)} \right), \quad (7)$$

for class embedding function e , input images \mathbf{x}_i with corresponding class labels y_i , indicator function $\mathbb{1}$, and scalar t .

2.4 Multi-generator Methods

Many generative style transfer approaches use multiple generators, with a pair of generators for each pair of domains, one transferring domain A to domain B and the other transferring domain B to domain A. Examples of these approaches are shown in Fig. 3. Multi-generator methods are particularly common in unsupervised settings,

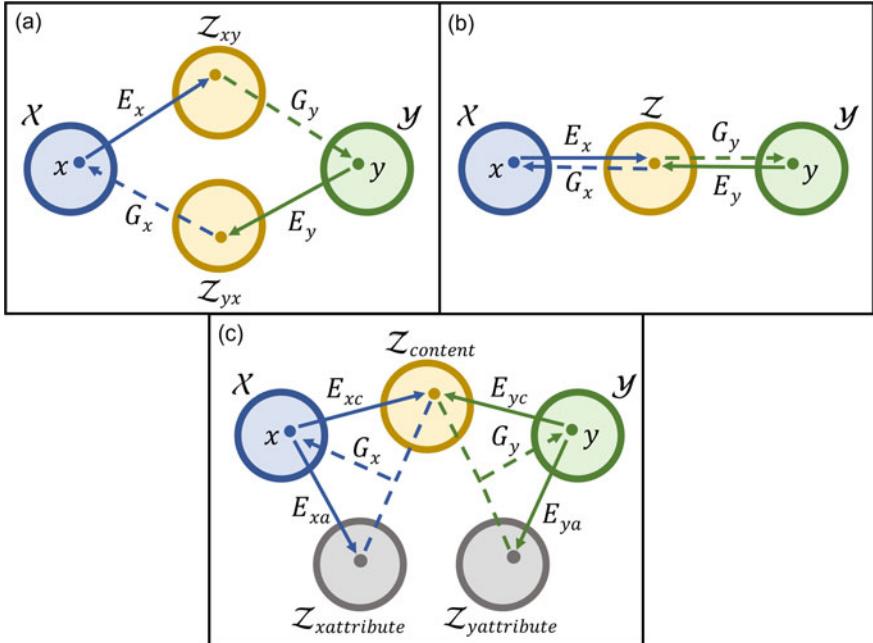


Fig. 3 Unsupervised image-to-image translation methods for images x, y from domains \mathcal{X}, \mathcal{Y} , with encoders E , generators G , and latent spaces \mathcal{Z} . **a** CycleGAN [63] uses domain-specific latent spaces, **b** UNIT [35] uses a shared latent space, **c** MUNIT [23] and DRIT [31] decompose the latent space into domain-specific attribute (style) spaces and a single shared content (structure) latent space. Diagram adapted from [32]

where there are no paired images available in different domains for training, making it more difficult to retain structural information in generated outputs.

The most commonly used unsupervised method is CycleGAN [63], a method whereby one GAN generates the stylised image, and a second GAN attempts to reconstruct the original image from the stylised image. A *cycle consistency loss* is used to measure the similarity between original and reconstructed images:

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{\mathbf{x} \sim p_{data(x)}}[||F(G(\mathbf{x})) - \mathbf{x}||_1] + \mathbb{E}_{\mathbf{y} \sim p_{data(y)}}[||G(F(\mathbf{y})) - \mathbf{y}||_1],$$

for generators G, F . This loss is minimised to reward the accurate reconstruction of original data. This ensures that the GANs are capable of generating varied data, avoiding the issue of *mode collapse*, where generated images all resemble the most common image in the modelled distribution. The CycleGAN loss is a combination of the two GAN-specific losses and the cycle consistency loss:

$$\mathcal{L}_{CycleGAN}(G, F, D_G, D_F) = \mathcal{L}_{GAN}(G, D_G) + \mathcal{L}_{GAN}(F, D_F) + \lambda \mathcal{L}_{cyc}(G, F),$$

where $\lambda \geq 0$ controls the relative importance of the different losses.

A different approach to combining two generators is taken in unsupervised image-to-image translation (UNIT) [35], with each of the domain-specific GANs sharing a latent space to learn the joint distribution of the domains instead of separately learning the marginal distributions. Diverse image-to-image translation via disentangled representations (DRIT) [31] is a hybrid between the CycleGAN and UNIT approaches, with the latent space being decomposed into a shared part and domain-specific parts. The shared part is the *content space*, which captures structure, and the domain-specific parts are the *attribute spaces*, which separately capture the style of each domain. The content encoding of an image can be combined with a style encoding from a different domain to perform style transfer. This approach uses a cross-cycle consistency loss, a multi-domain extension of the cycle consistency loss, which only applies to a single pair of domains. When reconstructing the original version of an image from any style-transferred version, the generator simply switches the style embedding and leaves the content embedding unchanged. Multimodal unsupervised image-to-image translation (MUNIT) [23] works similarly to DRIT, but instead of concatenating the style and content embeddings, it uses adaptive instance normalisation to combine style and content embeddings, and instead of cross-cycle consistency loss, MUNIT uses latent reconstruction losses.

3 Stain Normalisation

To overcome the domain differences in histopathology datasets, researchers may use style transfer methods to normalise digital pathology images. Normalising training data reduces variability, which can improve convergence speed, reduce overfitting to extreme colour values, and reduce confounding effects from stain variation. Normalising during inference reduces the domain gap between in-distribution and out-of-distribution data, making it more likely that the trained model will correctly interpret the image. Stain normalisation may also have a direct clinical benefit, with the increased consistency of normalised images allowing pathologists to make diagnoses faster and with greater confidence [48].

3.1 Traditional Normalisation

Histopathological stain normalisation was already a commonly researched task before the rise of deep learning, with a variety of statistical and spectral matching techniques used [56]. Many of these approaches used colour deconvolution to separate stains, and then normalised each independently. Four traditional approaches have remained particularly prevalent in deep learning era research, each named after their primary author—Reinhard [46], Macenko [37], Khan [28], and Vahadane [58] normalisation.

Reinhard normalisation [46] was the earliest of these approaches to be developed and was the only one of the four to not have been specifically developed for use in histopathology. It is a standard statistical normalisation approach (in that it works by subtracting a mean and scaling by a variance), that works in perception-based $l\alpha\beta$ colour space, for radiance channel l , blue-yellow channel α , and red-green channel β . This colour space has reduced correlation between colour channels compared to the standard red, green, and blue (RGB) space, improving the normalisation.

Macenko normalisation [37] also converts images to a different colour space, the optical density space, a logarithmic version of the RGB space. In optical density space, stains can be linearly combined, allowing for easier separation/combination of stains. Colour deconvolution is performed using singular value decomposition to separate stain vectors from saturation values, then an estimate of the maximum intensity of each stain is then generated and used to scale all stains to have the same maximum intensity.

Unlike the previous approaches, Khan normalisation [28] is a trainable approach which requires a target domain image as well as the source image to be normalised. It uses a trainable colour deconvolution where colour-based relevance vector machine classifiers generate stain matrices for both the input and target domain image. Then, to normalise the input image, a non-linear mapping is applied to each colour channel separately based on the corresponding colour channel statistics in the target domain image.

Vahadane normalisation [58] is also a trainable approach which requires a target domain image. Both the source and target images undergo colour deconvolution in optical density space to separate stains using a sparse non-negative matrix factorization, which was found to be more robust to uneven stain proportions than singular value decomposition. The deconvolution approach is iteratively optimized using sparse coding and dictionary learning to generate a stain colour appearance matrix and a stain density matrix for each image, where the stain density matrix captures the locations where each stain is present, and the colour appearance matrix captures the chromatic properties of each stain. The stain density of the source is scaled using a similar approach to the intensity scaling in Macenko normalisation and then combined with the colour appearance of the target to generate a normalised source image. By retaining the stain density of the source image, this approach attempts to maintain the structure of the original image.

These traditional approaches are often beneficial to downstream AI performance when using external data [26, 47, 50]. However, these approaches all have weaknesses and imperfections, for example, Reinhard normalisation often applies stains to background areas [28, 57, 58]. Macenko normalisation is prone to generating artifacts [28] and applying stains to the wrong regions [57]. Both Macenko and Reinhard normalisations can lose structural information [58]. Khan normalisation does not reproduce the less abundant stain(s) as well as the most abundant [58] and can incorrectly alter the colour of dyes [49, 57]. Vahadane and Macenko normalisations both struggle to separate larger numbers of stains [26]. Khan and Vahadane normalisation both use a single target image, requiring careful selection to ensure the target image is representative of the target domain as the normalised image can

vary greatly based on the target image. Despite the potential issues, these are all very popular normalisation approaches which typically offer greater benefits than using non-normalised histopathology images in artificial intelligence.

4 Generative Stain Normalisation

In recent years, the field of deep learning has rapidly grown, with a wide array of new techniques being developed. Many recently developed stain normalisation approaches use deep learning generative AI methods. The most common model in this field is the GAN, with methods either using a single generator in a supervised setting or a pair of generators in an unsupervised setting.

4.1 Single Generator Normalisation Approaches

Stain-style transfer (SST) [13] is a supervised stain normalisation approach which is trained to apply stain colours to the greyscale version of an input image. This is trained using a standard GAN adversarial loss, a Euclidean distance reconstruction loss, and a Kullback-Leibler divergence feature-preserving loss. The method was evaluated for the classification of lymph node metastasis [7], with slides from one institution used for training and validation, and slides from another used for testing. The proposed style transfer approach outperformed four non-GAN normalisation approaches, including Macenko and Reinhard normalisation, but did not fully overcome the domain gap, with an AUC of 0.92 compared to the target AUC of 0.98, obtained by training and testing on images from the same institution.

This approach was iterated on by the SSIM-GAN and DSCSI-GAN methods [34], where the reconstruction losses were based on SSIM and directional statistics-based colour similarity index (DSCSI), respectively. These reconstruction losses are more focused than the Euclidean loss, which disincentivises any change between the input and generated image. These models had slightly improved performance over SST but were still far from completely overcoming the domain gap, with SSIM-GAN and DCSGI-GAN achieving AUCs of 0.90 and 0.91, compared to a target of 0.97.

A similar approach [8] used a morphological reconstruction loss based on the colour gradient field. In combination with a GAN adversarial loss and a cross-entropy classification loss, this approach was beneficial in multiple tasks (three classifications, one segmentation, on a total of three datasets), including outperforming Khan normalisation. Multiple sources of stain variability were investigated, with one test set digitised using a different scanner and one sourced from different pathology centres. However, this approach also did not fully bridge the domain gaps.

Contrastive Learning for Unpaired Image-to-Image Translation (CUT) [44] is an approach which was not developed for stain normalisation but which has been applied to the task in subsequent research [2, 64]. A *patchwise contrastive loss* is used to

make patch embeddings similar for patches that are drawn from the same location in the real and generated image, and dissimilar for patches drawn from different locations. The default version of the model uses an adversarial loss and an identity loss as well as the patchwise contrastive loss, with the faster version, FastCUT, not including the identity loss. The authors described FastCUT as a one-sided version of CycleGAN, allowing for much faster training with much lower memory requirements due to only using a single GAN rather than the two GANs employed in CycleGAN.

Stain-to-Stain Translation (STST) [47] is an approach which builds on pix2pix [24], a supervised conditional GAN (cGAN) approach with an L1 reconstruction loss. The model was used to normalise data from one scanner to appear as though it was scanned using a different scanner, and was found to be beneficial to a greater extent than Reinhard, Macenko, Khan, and Vahadane normalisations using 10 metrics including SSIM and mean squared error, while running faster than any of the other approaches. However, the effects of this approach on any downstream AI task were not analysed.

Colour adaptive GAN (CAGAN) [14] is another greyscale-supervised pix2pix-based approach which aims to make stain normalisation more consistent. The proposed consistency approach uses two decoders (within the generator) with different structures to generate two stain-normalised versions of the same input image. Perturbations (augmentations) are applied to input images and to feature-space representations separately for each decoder, and the model is trained in a self-supervised fashion to overcome these perturbations, using a mean distance consistency loss between the two versions of the generated image. These generated images are also compared to the input images using a content loss between the feature representations for the target domain, and a histogram loss based on the Hellinger distance for the source domain. These losses are combined with an adversarial loss and an L1 supervised loss between real and generated images. This complex combination of losses was found to be beneficial, with CAGAN reported to outperform traditional normalisation, STST, a non-generative network-based approach [55], and the CycleGAN-based StainGAN for breast cancer classification and mutation status classification using glioma data. It is worth noting that all of these approaches provided large improvements over having no stain normalisation for most metrics. For five out of six metrics, CAGAN for stain normalisation also outperformed a stain augmentation approach which was previously reported to be better than a range of stain normalisation across various downstream tasks [55].

4.2 Multi-generator Normalisation Approaches

Standard CycleGAN

The most commonly used multi-generator approach in histopathology normalisation is the CycleGAN [63]. In our previous research [11], we evaluated normalisation approaches in the context of mitosis detection [4], and found that CycleGAN was only beneficial with one of our two baseline methods, with this being a marginal

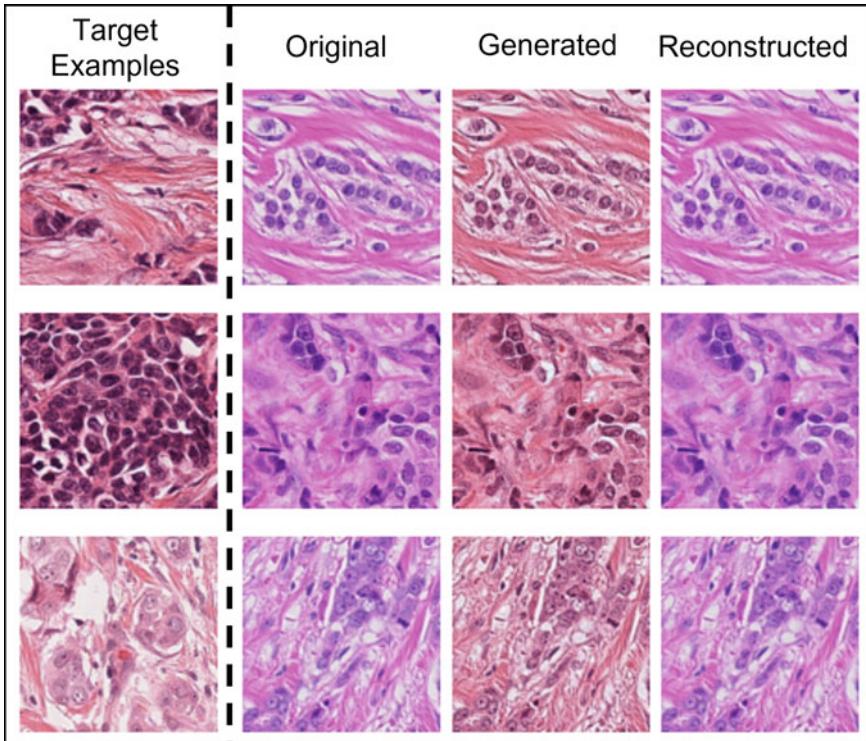


Fig. 4 CycleGAN stain normalisation applied to breast cancer tissue, with a different scanner used in the original and target domains. Figure adapted from [11]

benefit. This indicates that CycleGAN typically generated images which were too artificial to be interpreted more accurately by the trained detection models. Despite this, CycleGAN-generated images visually appeared to be high quality, as shown in Fig. 4. Macenko normalisation was not beneficial for either baseline method.

CycleGAN performance has also been found to be inconsistent in other studies. It was found to be less beneficial than Vahadane normalisation for lymph node cancer classifications in breast tissue and colon tissue in one study including two different classifiers [52]. A different study found CycleGAN to be more beneficial than a non-generative normalisation approach [6] in the classification of cancer in prostate biopsies with two independent test sets [54]. It has also been found to be beneficial compared to Macenko and Vahadane normalisations in ovarian cancer subtype classification when using external test data [51]. In each of these studies, CycleGAN gave an improvement over the baseline with no normalisation, though in our study it only provided a benefit for one of the two baseline approaches [11].

While CycleGAN generates visually pleasing images, it has some limitations. In the context of renal glomeruli segmentation with different stains, it was found that CycleGAN-generated images belong to distinct, variable distributions, and that this

has profound effects on the downstream task. It was also shown how CycleGAN can “hallucinate”, inserting new objects which do not fit the original structure [16], though this can be somewhat mitigated through the inclusion of normalisation layers in the network [59]. CycleGAN is computationally taxing due to requiring two GANs, which is especially problematic when using data from several domains, where a different CycleGAN model needs to be trained for each pair of domains.

Non-standard CycleGAN

CycleGAN can be adjusted in various ways to address the limitations, with one study finding that “changing the generator model to a smaller U-net-like architecture, adding an identity loss term, increasing the batch size and the learning all led to improved training stability and performance” [17], which was assessed using the structural similarity index (SSIM). The identity loss was an L1 distance between the input and output of each generator to discourage large changes in the generated images, and the weight of this loss was decreased during training until it reached zero, meaning the identity loss was only used to stabilise training and avoid local optima. This adjusted CycleGAN was found to be beneficial in the context of segmenting IHC-stained renal tissue from multiple centres.

The same authors iterated on their previous approach to generate Residual CycleGAN [16] which added skip-connections from the original image to the output of the generator, focusing model training on adjusting colours rather than on the entire regeneration of images. This was also beneficial, with the authors reporting that Residual CycleGAN gave better performance than the previous approach, as well as Reinhard, Macenko, and Vahadane normalisations, StainGAN, and a lookup table normalisation approach [6], in the context of colon tissue segmentation. It was also found that Residual CycleGAN had a reduced frequency of hallucination compared to the standard CycleGAN or StainGAN.

StainGAN [50] is an adjusted CycleGAN which uses a target style distribution rather than an individual target style image. This was found to improve breast cancer classification performance above that which is obtained by any of Reinhard, Macenko, Khan, or Vahadane normalisation, as well as improving generated image quality according to four similarity metrics. A transitive adversarial network (TAN) [12] has been proposed as a faster alternative to this approach, where instead of a single target style image, the discriminator is provided with a new, randomly sampled style image each time it is invoked. This was found to improve SSIM over StainGAN but was not evaluated for the downstream segmentation task.

The information learned in StainGAN can be leveraged much more efficiently using a teacher-student learning approach to distil knowledge from the CycleGAN-based StainGAN to a convolutional neural network, StainNet [26]. For this approach, the StainGAN output is treated as the ground truth for the input image, allowing StainNet to be trained in a fully supervised manner using an L1 loss between the StainGAN output and the StainNet output. StainNet was found to give similar SSIM scores and downstream classification scores to StainGAN for tasks in histopathology and cytopathology. StainNet can normalise a whole slide image (WSI) in 40 s

with much better structural similarity than StainGAN, which is prone to generating differing colours in neighbouring patches leading to an artificial checkerboard effect.

SegCN-Net [39] is another CycleGAN-based approach which uses self-supervised segmentation to preserve structural information. The model is trained with standard cycle consistency and adversarial losses, with additional segmentation losses which compare feature map similarities between real and generated images, and between real and reconstructed images, using mean squared error. This method was found to give better classification results for lymph node classification than standard normalisation approaches (Reinhard, Macenko, Vahadane), standard CycleGAN, and two previous CycleGAN adaptations [19, 62]. It also outperformed these CycleGAN adaptations for the segmentation of glands in colorectal samples and outperformed standard CycleGAN according to a structural similarity index.

Which normalisation method is best?

A recent comparison of stain normalisation techniques was performed in the context of colorectal cancer [2]. This included Reinhard, Macenko, Khan, and Vahadane normalisations, CycleGAN, CUT, FastCUT, and two other GAN-based approaches which were not originally developed for use in stain normalisation - Geometry-consistent GAN (GcGAN) [18] and AI-FFPE [43]. For one dataset, Macenko normalisation was the best approach according to SSIM and LPIPS, and FastCUT was best according to a pixel-level similarity metric (PSNR). For the other dataset, CycleGAN was the best for each of the three metrics. The generative normalisation approaches were also evaluated subjectively by pathologists, with CycleGAN found to generate the most realistic samples for both datasets. It was not clear which method was best when considering downstream classification—out of 8 accuracy measures across 4 experiments with the generative approaches, FastCUT was best three times, CycleGAN twice, GcGAN twice, and AI-FFPE once. Macenko normalisation outperformed all generative approaches for two of the eight measures, with the best generative approach being better than the best traditional approach for the others. Further, when considering mean precision, recall, and Dice scores, FastCUT was best in some evaluations and GcGAN in others, with generative approaches typically performing much better than traditional approaches by these metrics.

Another recent study [64] compared similar methods for stain-to-stain translation, a related task which is likely to provide useful insights for stain normalisation. The authors compared UNIT, MUNIT, pix2pix, CycleGAN, StainGAN, StainNet, CUT, Unsupervised content-preserving Transformation for Optical Microscopy (UTOM) [33], and Reinhard, Macenko, and Vahadane normalisations. The default CycleGAN performed best in terms of the FID, and pix2pix performed best in terms of SSIM. These metrics may miss important information when taken individually, and when accounting for both metrics, StainGAN and UTOM also performed well. All of the evaluated approaches outperformed Reinhard, Macenko, and Vahadane normalisations according to FID, indicating the benefit that GAN-based approaches can offer.

Another aspect of model selection is efficiency—many applications are resource-limited, meaning that computational complexity is an important factor. In the stain-to-stain translation study [64], StainNet and Reinhard normalisation were by far the fastest approaches in inference regardless of available hardware. GAN-based models

were particularly slow when a graphics processing unit (GPU) was unavailable, but were significantly sped up by the use of a GPU. Model training time was found to depend on the number of GANs used, with multi-GAN methods taking days to train, compared to hours for the non-generative and single GAN methods.

Overall, it is not clear which method of stain normalisation will be best in any given scenario, with different studies finding different methods to be best [2, 26, 47, 49, 50, 58, 61, 64]. A range of factors are likely to influence performance, including experimental design, evaluation approach, target domain image selection, stain type (H&E, IHC), and other image qualities (saturation, hue, etc.). Ideally, then, researchers should try many different stain normalisation approaches and select the one that performs best for the given task. This is unlikely to be practical for most researchers, so when selecting, two main factors can be considered to make informed decisions. First, the state-of-the-art generative approaches often, but not always, outperform traditional approaches [2, 26, 47, 50, 64]. Second, GAN-based methods have higher computational requirements than traditional normalisation approaches, with days of GPU time often needed for the training of multi-GAN methods, and these methods taking longer in inference regardless of available hardware. GAN-based methods are unlikely to be practical in situations where GPUs are not available.

5 Augmentation and Synthesis in Histopathology

Generative approaches in histopathology are not only used for stain normalisation but also for augmenting training data and synthesising new data. Stain augmentation is used during model training to artificially increase the heterogeneity of staining in the training dataset, increasing model robustness. This may give better performance than normalisation in some cases, for example, one study found that using a CycleGAN for augmentation provided slightly improved segmentation results over CycleGAN for normalisation using multi-centre kidney tissue images [9]. Synthesis is the creation of entirely new samples from the distribution of the training data, which is another approach to artificially increase the size of the training dataset. Despite the potential benefits of augmentation and synthesis, a review of GANs in histopathology up to March 2021 found that normalisation was the most popular approach [25].

6 Conclusion

The clinical utility of artificial intelligence in histopathology is limited by the variability of digital pathology data. It is often impractical or impossible to collect large enough repositories of varied data to train models to be truly robust to the many different sources of variability, leading to models generalising poorly on external datasets. Stain normalisation methods have been widely researched as an approach to reduce variability, from traditional statistics-based approaches to modern deep learning methods using generative AI. The most common generative approach is the

generative adversarial network (GAN), typically either a single GAN in a supervised setting or multiple GANs in an unsupervised setting. GAN-based stain normalisation is beneficial for many tasks, though computational complexity is much higher than for non-generative approaches. The performance of any specific stain normalisation method varies greatly in different scenarios, with it not being clear which is the best generative approach, and with non-generative approaches outperforming generative approaches in some cases.

Acknowledgements JB is supported by the UKRI Engineering and Physical Sciences Research Council (EPSRC) [EP/S024336/1]. For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising from this submission.

References

1. Ahmed, A. A., Abouzid, M., & Kaczmarek, E. (2022). Deep learning approaches in histopathology. *Cancers*, 14(21), 5264.
2. Altini, N., Marvulli, T. M., Zito, F. A., Caputo, M., Tommasi, S., Azzariti, A., Brunetti, A., Prencipe, B., Mattioli, E., De Summa, S., & et al. (2023). The role of unpaired image-to-image translation for stain color normalization in colorectal cancer histology classification. *Computer Methods and Programs in Biomedicine*, 107511.
3. Aubreville, M., Bertram, C. A., Breininger, K., Jabari, S., Stathonikos, N., & Veta, M. (2022). Mitosis domain generalization challenge 2022. In *25th International conference on medical image computing and computer assisted intervention (MICCAI)*.
4. Aubreville, M., Stathonikos, N., Bertram, C. A., Klopfleisch, R., Ter Hoeve, N., Ciompi, F., Wilm, F., Marzahl, C., Donovan, T. A., Maier, A., et al. (2023). Mitosis domain generalization in histopathology images—the MIDOG challenge. *Medical Image Analysis*, 84, 102699.
5. Baidoshvili, A., Bucur, A., van Leeuwen, J., van der Laak, J., Kluij, P., & van Diest, P. J. (2018). Evaluating the benefits of digital pathology implementation: Time savings in laboratory logistics. *Histopathology*, 73(5), 784–794.
6. Bejnordi, B. E., Litjens, G., Timofeeva, N., Otte-Höller, I., Homeyer, A., Karssemeijer, N., & Van Der Laak, J. A. (2015). Stain specific standardization of whole-slide histopathological images. *IEEE Transactions on Medical Imaging*, 35(2), 404–415.
7. Bejnordi, B. E., Veta, M., Van Diest, P. J., Van Ginneken, B., Karssemeijer, N., Litjens, G., Van Der Laak, J. A., Hermans, M., Manson, Q. F., Balkenhol, M., et al. (2017). Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA*, 318(22), 2199–2210.
8. BenTaieb, A., & Hamarneh, G. (2017). Adversarial stain transfer for histopathology image analysis. *IEEE Transactions on Medical Imaging*, 37(3), 792–802.
9. Bouteldja, N., Hölscher, D. L., Bülow, R. D., Roberts, I. S., Coppo, R., & Boor, P. (2022). Tackling stain variability using cyclegan-based stain augmentation. *Journal of Pathology Informatics*, 13, 100140.
10. Breen, J., Allen, K., Zucker, K., Adusumilli, P., Scarsbrook, A., Hall, G., Orsi, N. M., & Ravikumar, N. (2023). Artificial intelligence in ovarian cancer histopathology: A systematic review. *NPJ Precision Oncology*, 7(1), 83.
11. Breen, J., Zucker, K., Orsi, N. M., Ravikumar, N. (2021). Assessing domain adaptation techniques for mitosis detection in multi-scanner breast cancer histopathology images. In *Proceedings of the biomedical image registration, domain generalisation and out-of-distribution analysis: MICCAI 2021 challenges: MIDOG 2021, MOOD 2021, and Learn2Reg 2021, held*

- in conjunction with MICCAI 2021* (pp. 14–22), Strasbourg, France, September 27–October 1, 2021. Springer.
- 12. Cai, S., Xue, Y., Gao, Q., Du, M., Chen, G., Zhang, H., & Tong, T. (2019). Stain style transfer using transitive adversarial networks. In *Proceedings of the 2nd international workshop and machine learning for medical image reconstruction, MLMIR 2019, Held in Conjunction with MICCAI 2019* (pp. 163–172), Shenzhen, China, October 17, 2019. Springer.
 - 13. Cho, H., Lim, S., Choi, G., & Min, H. (2017). Neural stain-style transfer learning using GAN for histopathological images. arXiv preprint [arXiv:1710.08543](https://arxiv.org/abs/1710.08543)
 - 14. Cong, C., Liu, S., Di Ieva, A., Pagnucco, M., Berkovsky, S., & Song, Y. (2022). Colour adaptive generative networks for stain normalisation of histopathology images. *Medical Image Analysis*, 82, 102580.
 - 15. da Silva, L. M., Pereira, E. M., Salles, P. G., Godrich, R., Ceballos, R., Kunz, J. D., Casson, A., Viret, J., Chandarlapaty, S., Ferreira, C. G., et al. (2021). Independent real-world application of a clinical-grade automated prostate cancer detection system. *The Journal of Pathology*, 254(2), 147–158.
 - 16. de Bel, T., Bokhorst, J. M., van der Laak, J., & Litjens, G. (2021). Residual cycled GAN for robust domain transformation of histopathological tissue slides. *Medical Image Analysis*, 70, 102004.
 - 17. de Bel, T., Hermans, M., Kers, J., van der Laak, J., & Litjens, G. (2019). Stain-transforming cycle-consistent generative adversarial networks for improved segmentation of renal histopathology. In *International conference on medical imaging with deep learning* (pp. 151–163). PMLR.
 - 18. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Zhang, K., & Tao, D. (2019). Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2427–2436).
 - 19. Gadermayr, M., Appel, V., Klinkhammer, B. M., Boor, P., & Merhof, D. (2018). Which way round? A study on the performance of stain-translation for segmenting arbitrarily dyed histological images. In *Proceedings of the 21st international conference medical image computing and computer assisted intervention—MICCAI 2018* (pp. 165–173), Granada, Spain, September 16–20, 2018, Part II 11. Springer.
 - 20. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial networks. arXiv preprint [arXiv:1406.2661](https://arxiv.org/abs/1406.2661)
 - 21. Hanna, T. P., King, W. D., Thibodeau, S., Jalink, M., Paulin, G. A., Harvey-Jones, E., O’Sullivan, D. E., Booth, C. M., Sullivan, R., & Aggarwal, A. (2020). Mortality due to cancer treatment delay: Systematic review and meta-analysis. *BMJ*, 371.
 - 22. Hanna, M. G., Reuter, V. E., Samboy, J., England, C., Corsale, L., Fine, S. W., Agaram, N. P., Stamatos, E., Yagi, Y., Hameed, M., et al. (2019). Implementation of digital pathology offers clinical and operational increase in efficiency and cost savings. *Archives of Pathology and Laboratory Medicine*, 143(12), 1545–1555.
 - 23. Huang, X., Liu, M. Y., Belongie, S., & Kautz, J. (2018). Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 172–189).
 - 24. Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125–1134).
 - 25. Jose, L., Liu, S., Russo, C., Nadort, A., & Di Ieva, A. (2021). Generative adversarial networks in digital pathology and histopathological image processing: A review. *Journal of Pathology Informatics*, 12(1), 43.
 - 26. Kang, H., Luo, D., Feng, W., Zeng, S., Quan, T., Hu, J., & Liu, X. (2021). StainNet: A fast and robust stain normalization network. *Frontiers in Medicine*, 8, 746307.
 - 27. Kang, M., & Park, J. (2020). ContraGAN: Contrastive learning for conditional image generation. *Advances in Neural Information Processing Systems*, 33, 21357–21369.
 - 28. Khan, A. M., Rajpoot, N., Treanor, D., & Magee, D. (2014). A nonlinear mapping approach to stain normalization in digital histopathology images using image-specific color deconvolution. *IEEE Transactions on Biomedical Engineering*, 61(6), 1729–1738.

29. Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint [arXiv:1312.6114](https://arxiv.org/abs/1312.6114)
30. Larsen, A. B. L., Sønderby, S. K., Larochelle, H., & Winther, O. (2016). Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning* (pp. 1558–1566). PMLR.
31. Lee, H. Y., Tseng, H. Y., Huang, J. B., Singh, M., & Yang, M. H. (2018). Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 35–51).
32. Lee, H. Y., Tseng, H. Y., Mao, Q., Huang, J. B., Lu, Y. D., Singh, M., & Yang, M. H. (2020). DRIT++: Diverse image-to-image translation via disentangled representations. *International Journal of Computer Vision*, 128, 2402–2417.
33. Li, X., Zhang, G., Qiao, H., Bao, F., Deng, Y., Wu, J., He, Y., Yun, J., Lin, X., Xie, H., & et al. (2021). Unsupervised content-preserving transformation for optical microscopy. *Light: Science and Applications*, 10(1), 44.
34. Liang, H., Plataniotis, K. N., & Li, X. (2020). Stain style transfer of histopathology images via structure-preserved generative learning. In *Proceedings of the 3rd international workshop on machine learning for medical image reconstruction: MLMIR 2020, Held in Conjunction with MICCAI 2020* (pp. 153–162), Lima, Peru, October 8, 2020. Springer.
35. Liu, M. Y., Breuel, T., & Kautz, J. (2017). Unsupervised image-to-image translation networks. *Advances in Neural Information Processing Systems*, 30.
36. Lyon, H. O., De Leenheer, A., Horobin, R., Lambert, W., Schulte, E., Van Liedekerke, B., & Wittekind, D. (1994). Standardization of reagents and methods used in cytological and histological practice with emphasis on dyes, stains and chromogenic reagents. *The Histochemical Journal*, 26, 533–544.
37. Macenko, M., Niethammer, M., Marron, J. S., Borland, D., Woosley, J. T., Guan, X., Schmitt, C., & Thomas, N. E. (2009). A method for normalizing histology slides for quantitative analysis. In *2009 IEEE international symposium on biomedical imaging: from nano to macro* (pp. 1107–1110). IEEE.
38. Magee, D., Treanor, D., Crellin, D., Shires, M., Smith, K., Mohee, K., & Quirke, P. (2009). Colour normalisation in digital histopathology images. In *Proceedings of the optical tissue image analysis in microscopy, histopathology and endoscopy (MICCAI workshop)* (vol. 100, pp. 100–111). Daniel Elson London.
39. Mahapatra, D., Bozorgtabar, B., Thiran, J. P., & Shao, L. (2020). Structure preserving stain normalization of histopathology images using self supervised semantic guidance. In *Proceedings of the 23rd international conference on medical image computing and computer assisted intervention—MICCAI 2020* (pp. 309–319), Lima, Peru, October 4–8, 2020, Part V 23. Springer.
40. Michielli, N., Caputo, A., Scotto, M., Mogetta, A., Pennisi, O. A. M., Molinari, F., Balmativalo, D., Bosco, M., Gambella, A., Metovic, J., et al. (2022). Stain normalization in digital pathology: Clinical multi-center evaluation of image quality. *Journal of Pathology Informatics*, 13, 100145.
41. Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint [arXiv:1411.1784](https://arxiv.org/abs/1411.1784)
42. Mukhopadhyay, S., Feldman, M. D., Abels, E., Ashfaq, R., Beltaifa, S., Cacciabeve, N. G., Cathro, H. P., Cheng, L., Cooper, K., Dickey, G. E., et al. (2018). Whole slide imaging versus microscopy for primary diagnosis in surgical pathology: A multicenter blinded randomized noninferiority study of 1992 cases (pivotal study). *The American Journal of Surgical Pathology*, 42(1), 39.
43. Ozyoruk, K. B., Can, S., Darbaz, B., Başak, K., Demir, D., Gokceler, G. I., Serin, G., Hacisalihoglu, U. P., Kurtuluş, E., Lu, M. Y., & et al. (2022). A deep-learning model for transforming the style of tissue images from cryosectioned to formalin-fixed and paraffin-embedded. *Nature Biomedical Engineering*, 1–13.
44. Park, T., Efros, A. A., Zhang, R., Zhu, J. Y. (2020). Contrastive learning for unpaired image-to-image translation. In *Proceedings of the 16th European conference on computer vision—ECCV 2020* (pp. 319–345), Glasgow, UK, August 23–28, 2020, Part IX 16. Springer.

45. Rajaganesan, S., Kumar, R., Rao, V., Pai, T., Mittal, N., Sahay, A., Menon, S., & Desai, S. (2021). Comparative assessment of digital pathology systems for primary diagnosis. *Journal of Pathology Informatics*, 12(1), 25.
46. Reinhard, E., Adhikhmin, M., Gooch, B., & Shirley, P. (2001). Color transfer between images. *IEEE Computer graphics and applications*, 21(5), 34–41.
47. Salehi, P., & Chalechale, A. (2020). Pix2pix-based stain-to-stain translation: A solution for robust stain normalization in histopathology images analysis. In *2020 International conference on machine vision and image processing (MVIP)* (pp. 1–7). IEEE.
48. Salvi, M., Caputo, A., Balmativala, D., Scotto, M., Pennisi, O., Michielli, N., Mogetta, A., Molinari, F., & Fraggetta, F. (2023). Impact of stain normalization on pathologist assessment of prostate cancer: A comparative study. *Cancers*, 15(5), 1503.
49. Sethi, A., Sha, L., Vahadane, A. R., Deaton, R. J., Kumar, N., Macias, V., & Gann, P. H. (2016). Empirical comparison of color normalization methods for epithelial-stromal classification in h and e images. *Journal of pathology informatics*, 7(1), 17.
50. Shaban, M. T., Baur, C., Navab, N., & Albarqouni, S. (2019) Staingan: Stain style transfer for digital histological images. In *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019)* (pp. 953–956). IEEE.
51. Shin, S. J., You, S. C., Jeon, H., Jung, J. W., An, M. H., Park, R. W., & Roh, J. (2021). Style transfer strategy for developing a generalizable deep learning application in digital pathology. *Computer Methods and Programs in Biomedicine*, 198, 105815.
52. Stacke, K., Eilertsen, G., Unger, J., & Lundström, C. (2020). Measuring domain shift for deep learning in histopathology. *IEEE Journal of Biomedical and Health Informatics*, 25(2), 325–336.
53. Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021). Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, 71(3), 209–249.
54. Swiderska-Chadaj, Z., de Bel, T., Blanchet, L., Baidoshvili, A., Vossen, D., van der Laak, J., & Litjens, G. (2020). Impact of rescanning and normalization on convolutional neural network performance in multi-center, whole-slide classification of prostate cancer. *Scientific Reports*, 10(1), 1–14.
55. Tellez, D., Litjens, G., Bárdi, P., Bulten, W., Bokhorst, J. M., Ciompi, F., & Van Der Laak, J. (2019). Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *Medical Image Analysis*, 58, 101544.
56. Tosta, T. A. A., de Faria, P. R., Neves, L. A., & do Nascimento, M. Z. (2019). Computational normalization of h&e-stained histological images: Progress, challenges and future potential. *Artificial Intelligence in Medicine*, 95, 118–132.
57. Tosta, T. A. A., Freitas, A. D., de Faria, P. R., Neves, L. A., Martins, A. S., & do Nascimento, M. Z. (2023). A stain color normalization with robust dictionary learning for breast cancer histological images processing. *Biomedical Signal Processing and Control*, 85, 104978.
58. Vahadane, A., Peng, T., Sethi, A., Albarqouni, S., Wang, L., Baust, M., Steiger, K., Schlitter, A. M., Esposito, I., & Navab, N. (2016). Structure-preserving color normalization and sparse stain separation for histological images. *IEEE Transactions on Medical Imaging*, 35(8), 1962–1971.
59. Vasiljević, J., Nisar, Z., Feuerhake, F., Wemmert, C., & Lampert, T. (2022). Cyclegan for virtual stain transfer: Is seeing really believing? *Artificial Intelligence in Medicine*, 133, 102420.
60. Wilson, M. L., Fleming, K. A., Kuti, M. A., Looi, L. M., Lago, N., & Ru, K. (2018). Access to pathology and laboratory medicine services: A crucial gap. *The Lancet*, 391(10133), 1927–1938.
61. Zanjani, F. G., Zinger, S., Bejnordi, B. E., van der Laak, J. A., & et al. (2018). Histopathology stain-color normalization using deep generative models. *Medical Imaging with Deep Learning*.
62. Zhou, N., Cai, D., Han, X., & Yao, J. (2019). Enhanced cycle-consistent generative adversarial network for color normalization of h&e stained images. In *Proceedings 22nd international conference on medical image computing and computer assisted intervention—MICCAI 2019* (pp. 694–702), Shenzhen, China, October 13–17, 2019, Part I 22. Springer.

63. Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE international conference on computer vision (ICCV)*.
64. Zingman, I., Frayle, S., Tankoyeu, I., Sukhanov, S., & Heinemann, F. (2023). A comparative evaluation of image-to-image translation methods for stain transfer in histopathology. arXiv preprint [arXiv:2303.17009](https://arxiv.org/abs/2303.17009)

Augmenting Data from Epileptic Brain Seizures Using Deep Generative Networks



Jean-Philippe Thivierge

Abstract In many domains including medicine, biology, and neuroscience, rare events are the norm rather than the exception, limiting the ability to train intelligent systems to perform reliable pattern classification. This is the case when monitoring brain activity for epileptic seizures that constitute infrequent periods when abnormal electrical activity propagates across clusters of neurons. Here, as a solution, we describe how a generative adversarial network (GAN) can serve to produce synthetic examples that capture key features of epileptic activity observed in networks of *in vitro* cortical neurons. Further, GANs can generate novel patterns that deviate in systematic ways from the original data. A convolutional neural network whose goal was to classify healthy and seizure activity attained higher performance when trained on an augmented dataset composed of both original and synthetic data. Altogether, this work shows how GANs can provide data augmentation in a domain of epileptic seizures characterized by rare events.

Keywords Generative adversarial network · Epilepsy · Brain activity · Complexity

1 Introduction

Brain seizures are a state of activity where electrical impulses propagate across large clusters of neurons in a pathological fashion. The onset and origin of seizures are often difficult to predict yet are key to understanding their cause and develop treatments. Deep learning has been successful at detecting epilepsy [1]. Its performance, however, is limited by the fact that seizure events constitute rare occurrences in the brain and are difficult to identify [10, 16, 17, 24, 33].

Generative adversarial networks constitute a class of deep learning networks that have been fruitful at producing novel exemplars that closely resemble features of a

J.-P. Thivierge (✉)

School of Psychology and Brain and Mind Research Institute, University of Ottawa, Ottawa, Canada

e-mail: Jean-Philippe.Thivierge@uottawa.ca

given dataset [12, 14, 36]. In turn, these exemplars can be employed to perform *data augmentation* by creating a larger dataset than provided by the original exemplars. It is important to consider whether novel exemplars generated by GANs can capture important features of epileptic brain activity. One metric to describe this activity is its *complexity* [2, 15, 26, 28], reflecting the number of statistical dimensions needed to capture the variance in the data, as typically computed by eigenspectrum decomposition. Other metrics include the spatial and temporal correlations amongst neurons and the center of mass of propagating seizures, as discussed below.

In this chapter, we show how GANs can be employed to capture seizure activity in preparations of in vitro cortical neurons and exhibit many of the known features of this activity. This work is structured as follows: (i) we begin with a characterization of seizure activity recorded by a high-density multielectrode array (MEA) interfaced with cortical neurons; (ii) we describe a GAN network that is trained to generate novel exemplars that mimic brain seizures; (iii) we show how alterations to the GAN allow the network to explore a broad range of brain states from normal to pathological; and (iv) we examine the performance of a deep convolutional network trained to classify healthy and epileptic brain activity with data augmentation provided by the GAN network.

2 Characterization of Brain Seizures in Vitro

Activity from prefrontal cortex was recorded as previously described under application of a pro-epileptiform solution that promoted the emergence of seizures [4, 13, 18, 32, 34, 39, 40]. Brain activity was monitored by a high-density MEA that is sensitive to voltage deflections and comes in close contact with the surface of neurons. Activity was characterized by slow propagating waves that rotated in a spiral-like motion around a spatial locus called the center of mass (Fig. 1).

The center of mass of epileptic waves was computed based on the voltage of neural activity over time. Given a square matrix $A \in \mathbb{R}^{N \times N}$ where N are rows and columns of the data (Fig. 1), elements a_{ij} of the matrix are employed to compute the row

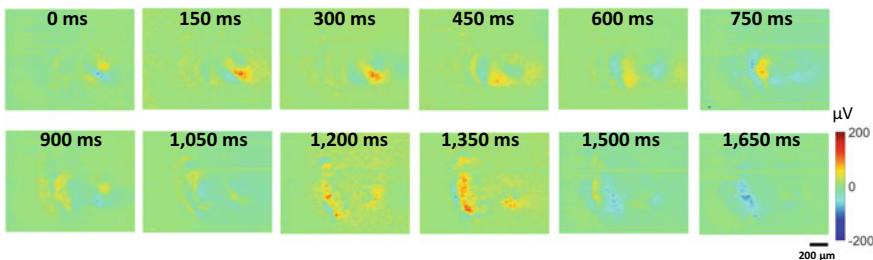


Fig. 1 Seizure activity characterized by spiral-like waves of cortical activity

$$r = \frac{\sum_{i,j}^N i \cdot a_{ij}}{\sum_{i,j}^N a_{ij}}, \quad (1)$$

and column

$$c = \frac{\sum_{i,j}^N j \cdot a_{ij}}{\sum_{i,j}^N a_{ij}}, \quad (2)$$

of the center of mass in space. The center of mass was highly consistent across epileptic waves recorded from the same clusters of neurons over time but varied across recording sessions with different clusters of neurons (Fig. 2). This indicates that the center of mass of epileptic waves is highly consistent over time, with individual waves following a stereotypical pattern of propagation across neurons.

Another key feature of seizure activity is that electrical impulses become highly correlated during epileptic waves. An example of correlations obtained during a single wave is shown in Fig. 3a. Correlations decrease rapidly as we move away from the center of mass. This feature reveals that correlations exhibit a specific spatial structure where neurons close to the center of mass are highly correlated while neurons farther away have almost no correlation.

Finally, seizure activity can be characterized by its *complexity*. This metric is estimated by first applying an eigenspectrum decomposition, yielding a set of ordered

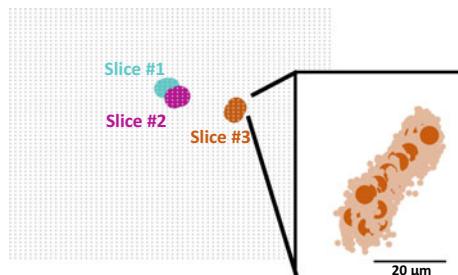


Fig. 2 Center of mass of seizures across three recordings. The inset is a zoom from one recording. Darker colors show the mean center of mass across time frames. Lighter colors are individual time frames. Background grey mesh shows the 64×64 grid of electrodes on the MEA

Fig. 3 **a** Correlation between the center of mass (shown by black circle) and the surrounding neural area.
b Correlations decrease as distance from the center of mass increases

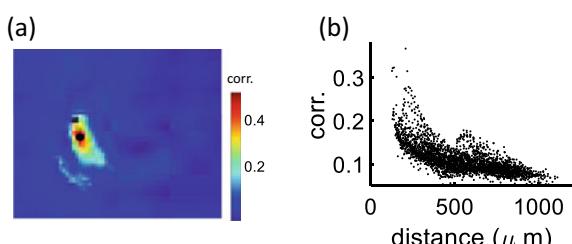
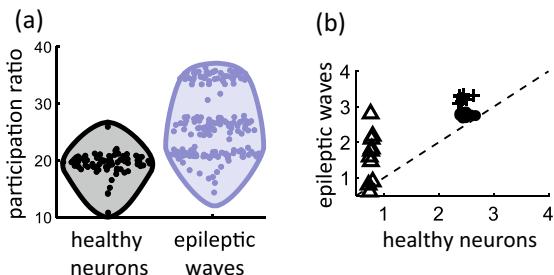


Fig. 4 Complexity of epileptic waves using the participation ratio (a) and LBMLE (b). In panel (b), the markers correspond to three different recordings of brain activity



eigenvalues $\lambda_1 > \lambda_2 > \dots > \lambda_N$ where N is the total number of dimensions considered. Complexity was then computed using the participation ratio [2, 15, 26, 28],

$$\text{PR} = \frac{\left(\sum_i^N \lambda_i\right)^2}{\sum_i^N \lambda_i^2}. \quad (3)$$

The PR is a ratio of the eigenspectrum's first and second moments. If seizures can be described by a small number of dimensions, the PR will be low. Conversely, a higher PR value is associated with a larger number of dimensions. More concretely, low dimensionality would be characterized by neurons exhibiting a simple pattern of activity, such as switching between an active and inactive state at regular intervals. Conversely, high dimensionality would arise when neurons produce intricate patterns of activity extending both spatially and over time.

The PR of epileptic waves was higher on average than healthy neurons (Fig. 4a), showing that seizures constitute a state of high brain complexity. Hence, the number of dimensions required to capture brain activity increases in epileptic waves relative to a base state of activity.

Because the participation ratio is inherently a linear metric, it is known to overestimate complexity [2]. To compensate for this effect, we examined a non-linear measure termed the Levina-Bickel maximum likelihood estimation (LBMLE) [25]. This measure employs a geometric approach to estimate the dimensionality of a set of data points. Applying LBMLE to three recordings of epileptic waves shows higher complexity than healthy neurons, corroborating the results obtained with the PR ratio (Fig. 4b). Further, the lower complexity values associated with LBMLE compared to PR are comparable to those obtained in related work [2].

In sum, we have characterized several features of seizure activity, including the center of mass, correlations, and complexity. The next section will examine to what extent these features can be captured by a deep GAN model that is trained to generate novel exemplars of seizure activity based on a small dataset of brain recordings.

3 Capturing Epileptic Waves with a Deep GAN Model

In GANs, two artificial networks are trained in a competitive fashion against one another [12]. One network, the “generator model” (G), aims to produce novel exemplars that closely capture the original data. The second network, the “discriminator model” (D), attempts to discriminate these exemplars from the real data. By training both networks concurrently, the generator network becomes increasingly more proficient at producing synthetic exemplars that evade the discriminator network.

The structure of the GAN is described as follows (Fig. 5a). A more detailed description is available in related work [5]. Briefly, the generator network contained six hidden layers each composed of rectified linear units (ReLU). The output layer was comprised of hypertan (htan) units. For the discriminator network, eight hidden layers with ReLU units were employed, along with htan units at the output layer. Each layer was preceded by a convolution step.

Training data was prepared by converting epileptic waves to images that captured the activation of neurons at fixed time points (Fig. 1). The complete set of training data was composed of 1,314 images extracted from 219 epileptic waves.

The training data is denoted as $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$, where \mathbb{P}_r is the observed distribution of the data. The goal pursued by GAN is to produce novel exemplars \tilde{x} whose distribution \mathbb{P}_g closely approximates \mathbb{P}_r . This can be performed by inputting noise to the generator network, $\tilde{x} = G_\theta(z)$, with noisy priors given by $\{z^{(i)}\}_{i=1}^m \sim p(z)$, where the input z is sampled from a Gaussian distribution.

A minimax objective function was employed to train both the generator and discriminator networks,

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_{z(z)}} [\log(1 - D(G(z)))] \quad (4)$$

where $V(D, G)$ is a min–max function and x corresponds to the observed data. The Adam optimizer [22] was employed to optimize this objective function, using a

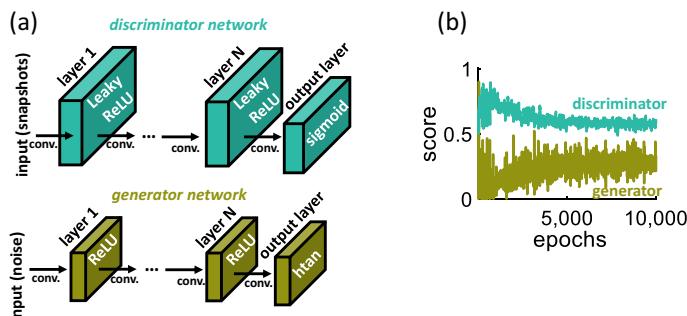


Fig. 5 **a** Architecture of the discriminator and generator networks. **b** Performance score over the course of training

discriminator learning rate of $\alpha = 0.0002$ and a generator learning rate of $\alpha = 0.001$. During training, the performance of the generator was evaluated with the score

$$S_G = \text{mean}(\hat{Y}_{\text{generated}}), \quad (5)$$

where $\hat{Y}_{\text{generated}}$ is the probability associated with the synthetic images. The performance score of the discriminator network was given by

$$S_D = 0.5 \text{ mean}(\hat{Y}_{\text{real}}) + 0.5 \text{ mean}(1 - \hat{Y}_{\text{generated}}), \quad (6)$$

where \hat{Y}_{real} is the output probability of the original images. Performance scores of the generator and discriminator networks began to saturate after approximately 10,000 training epochs (Fig. 5b). In a typical scenario, the generator and discriminator scores should be close to 0.5. This scenario, however, is not a requirement for a successful GAN, as several complementary measures can be employed to examine the ability of the model to produce faithful synthetic data.

After training the GAN model, a dataset of 1000 synthetic waves were generated by injecting random Gaussian-distributed input to the generator network. An examination of the synthetic epileptic waves produced by the GAN revealed similarities with the original data. Specifically, the synthetic waves showed regions where activity was high (Fig. 6a, in red) surrounded by regions with low activity (in green). Further, regions of high activity were clustered in space as in the neural data.

The correlation across locations on the synthetic images decreased with spatial distance (Fig. 6b). This is largely due to the spatial clustering of activity in the images. This result captures an important feature of neural data where the correlation between neurons was highest at nearby spatial locations (Fig. 3).

Another feature of the neural data is that epileptic waves exhibit a tight center of mass, meaning that activity is focused on a well-defined point in space. A similar

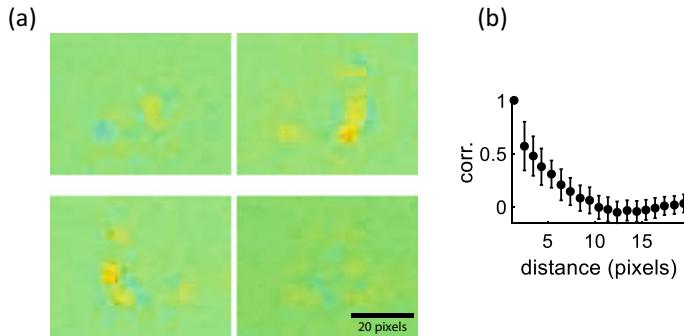
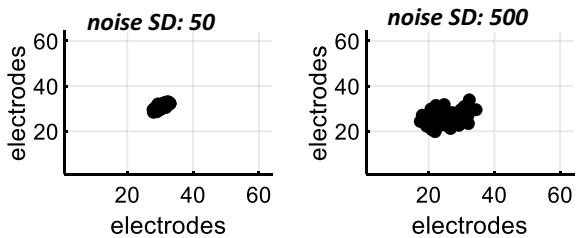


Fig. 6 **a** Synthetic epileptic waves generated by the GAN model. **b** Correlation between pixels relative to their spatial distance on the synthetic images. Vertical bars are standard error of the mean

Fig. 7 Center of mass of synthetic images with lower (SD: 50) or higher (SD: 500) noise in the input injected after training



effect is found with synthetic images (Fig. 7). Crucially, the degree of dispersion of the center of mass depends upon the statistics of the Gaussian input injected to the generator network after training. When the standard deviation (SD) of this input was increased by one order of magnitude from 50 to 500, the center of mass became visibly more diffuse. This result shows that the statistics of the input to the GAN model can be adjusted to explore a variety of images that deviate systematically from the observed brain data. Here, a more diffuse center of mass is indicative of weaker epileptic waves that do not occupy a well delimited region of space.

Finally, we examined the ability of the GAN model to produce images whose complexity approximated the seizure activity. A key factor in controlling the complexity of synthetic images was the mean of the Gaussian input injected after training. A close approximation of the complexity of epileptic waves, measured by the participation ratio, was obtained with a mean Gaussian input near zero (Fig. 8a). This point corresponds to the highest complexity value produced by the GAN model. Complexity decreased gradually when either increasing (positive values) or decreasing (negative values) the mean of the input. By adjusting the mean input, the GAN was able to capture a spectrum of complexity values ranging from epileptic waves to healthy neurons.

The performance of GANs relative to the mean of the Gaussian input was assessed by the Inception score (IS) [36]. This measure provides an estimate of the goodness-of-fit between synthetic and real data. The log of the IS was computed for ease of visualization (Fig. 8b); the lower this measure, the better the fit. A mean input near zero provided the highest fit to epileptic waves. Away from that value, the fit diminished rapidly. Thus, adjusting the mean input of the GAN allowed the model not only to capture the complexity of the observed data, but also to control the overall goodness-of-fit between epileptic waves and synthetic data.

Fig. 8 **a** The participation ratio (PR) of synthetic images shows a broad range of values that depend upon the mean of the input to the GAN model. **b** Inception score

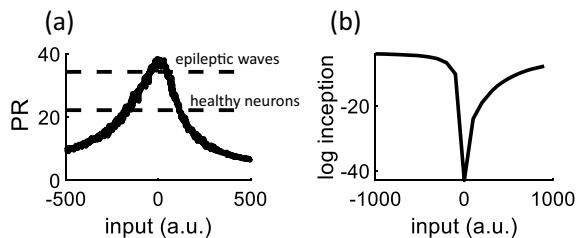
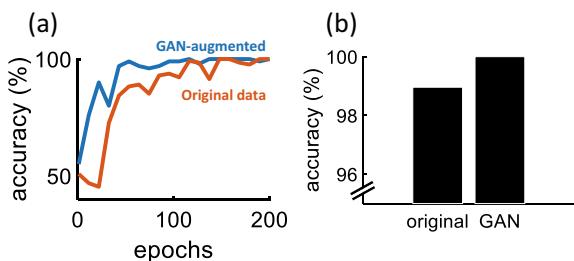


Fig. 9 **a** Training accuracy of CNN trained on original or GAN-augmented data.
b Testing accuracy



4 Training a Convolutional Network on Augmented Seizure Data

Next, we considered a proof-of-principle showing that the synthetic examples generated by the GAN model could improve the performance of a convolutional neural network (CNN) trained to classify healthy and epileptic activity. The network architecture consisted of seven layers including, in order: an input layer, a convolution layer, a ReLU layer, a max pooling layer, a fully connected layer, a softmax layer, and a classification layer. Default parameters from the Matlab Neural Networks toolbox (Mathworks, Natick) were employed. The learning rate was set to 0.0001. A set of training data was composed of 630 images of epileptic waves and an equivalent number of images from healthy neurons. Following training, a testing set composed of 684 images was employed to evaluate the classification capacity of the network. The performance of the CNN on these data was compared to an augmented data where 630 GAN-generated synthetic images of epileptic and healthy neurons were added to the data, effectively doubling the number of training images.

Both training and testing performance improved in the augmented dataset (Fig. 9a, b). The addition of synthetic examples increased the classification accuracy of healthy and seizure images. While testing accuracy was relatively high in the dataset prior to augmentation (98.97%), the addition of GAN-generated images improved it further (99.91%).

These results suggest that, in domains such as epileptic seizures where data is limited, data augmentation by GAN networks may improve classification performance. Of course, this improvement in performance may only be attained when the GAN generates novel exemplars that closely capture the original data.

5 Discussion

The current work described an application of a GAN network to generate synthetic data that closely approximated the statistics of brain seizures recorded in cortical neurons [5]. Synthetic images produced by the GAN captured the center of mass, spatial correlations, and complexity of epileptic waves. Going further, adjusting the

statistics of the Gaussian input delivered to the GAN after training allowed the model to capture a broad range of complexity values, ranging from healthy to pathological data. A deep CNN trained to classify brain activity benefited from augmented data that combined synthetic and real images of healthy and pathological neurons.

GAN networks trained to produce pathological brain activity can be employed for practical applications. First, the recent advent of neurostimulation allows physicians to apply electrical currents to patients with the aim of suppressing seizures [37]. The question that remains largely unanswered, however, is what specific stimulation protocols are optimal for seizure suppression [41]. In this respect, GAN networks can not only generate vast amounts of realistic synthetic data to examine this issue, but also provide data that span a broad range of states from healthy to pathological brain activity.

A second practical application of GANs is to generate augmented datasets aimed at training intelligent systems such as deep networks to classify brain activity and identify the spatial localization of seizures in patients [8, 19, 20]. GANs may be particularly useful in this respect given that seizure events are rare and therefore insufficient to train data-greedy models such as deep networks. Crucially, the contribution of our work is to show that GANs can replicate key features of brain seizures and may therefore serve as a reasonable proxy for data augmentation. Further, we provided a proof-of-principle that data augmentation can enhance the training and generalization performance of deep CNNs (Fig. 9).

Aside from the GAN model employed here, other generative approaches could serve to produce synthetic data [6, 21, 29, 35]. To date, however, few models have been implemented to capture brain activity [3, 27, 31, 38]. Aside from our own work [5], none thus far has been applied to epileptic seizures.

While our results demonstrate the ability of GANs to capture epileptic waves, a comparison with alternative models would be worth exploring. These models include diffusion models [9], variational autoencoders [23], and U-nets [7]. Diffusion models are based on likelihood estimation and in some instances can avoid the “data collapse” issue with GANs where the generated images are limited to a subset of the training space. A drawback, however, is that diffusion models are generally slower than GANs and require significant user intervention. Another form of generative model is variational autoencoders. This model reduces input data to a latent space of low dimensionality. The goodness-of-fit of generated images is typically inferior to GANs. Finally, U-nets are a generative model where original images are partitioned into basic constituents prior to reconstruction. In this model, the dimensions of the original and synthetic images must be identical. It remains unknown at present how these different approaches would compare when trained to produce brain seizure activity.

One drawback of GANs when applied to controlling seizures is the requirement that brain activity should be monitored in real time. Generative networks that can handle continuous data streams as input and produce synthetic data that unfolds over time is an active area of current research [11, 30]. Our current work should be viewed as a launchpad towards more sophisticated approaches that enable brain seizures to be monitored, replicated via synthetic data, and controlled in real time.

6 Conclusions

Epileptic seizures reflect not merely states of heightened activity but follow well-prescribed patterns of neural activation in the brain. A deep GAN successfully captured these patterns and was employed to show a range of complexity values spanning from healthy to pathological brain activity. This work opens avenues for employing GANs to generate augmented data sets of rare epileptic events that can serve to train intelligent systems that detect, predict, and localize seizures within the brain. Furthermore, GANs may accelerate the development of optimal neuro-stimulation protocols aimed at suppressing epileptic waves in clinical settings.

References

1. Abdelhameed, A., & Bayoumi, M. (2021). A Deep learning approach for automatic seizure detection in children with epilepsy. *Frontiers in Computational Neuroscience*, *15*, 650050. <https://doi.org/10.3389/fncom.2021.650050>
2. Altan, E., Solla, S. A., Miller, L. E., & Perreault, E. J. (2021). Estimating the dimensionality of the manifold underlying multi-electrode neural recordings. *PLoS Computational Biology*, *17*, e1008591. <https://doi.org/10.1371/journal.pcbi.1008591>
3. Arakaki, T., Barello, G., & Ahmadian, Y. (2017). *Capturing the diversity of biological tuning curves using generative adversarial networks*. arXiv preprint arXiv:170704582
4. Bear, J., & Lothman, E. W. (1993). An in vitro study of focal epileptogenesis in combined hippocampal-parahippocampal slices. *Epilepsy Research*, *14*, 183–193. [https://doi.org/10.1016/0920-1211\(93\)90043-7](https://doi.org/10.1016/0920-1211(93)90043-7)
5. Boucher-Routhier, M., & Thivierge, J.-P. (2023). A deep generative adversarial network capturing complex spiral waves in disinhibited circuits of the cerebral cortex. *BMC Neuroscience*, *24*, 22. <https://doi.org/10.1186/s12868-023-00792-6>
6. Brock, A., Donahue, J., & Simonyan, K. (2018). *Large scale GAN training for high fidelity natural image synthesis*. arXiv preprint arXiv:180911096
7. Chen, X., Li, Y., Yao, L., Adeli, E., & Zhang, Y. (2021). *Generative adversarial U-Net for domain-free medical image augmentation*. arXiv preprint arXiv:210104793
8. Chirasani, S. K. R., & Manikandan, S. (2022). A deep neural network for the classification of epileptic seizures using hierarchical attention mechanism. *Soft Computing*, *26*, 5389–5397. <https://doi.org/10.1007/s00500-022-07122-8>
9. Dhariwal, P., & Nichol, A. (2021). *Diffusion models beat GANs on image synthesis*
10. Dzhala, V. I., & Staley, K. J. (2003). Transition from interictal to ictal activity in limbic networks in vitro. *Journal of Neuroscience*, *23*, 7873–7880.
11. Esteban, C., Hyland, S. L., & Rätsch, G. (2017). *Real-valued (medical) time series generation with recurrent conditional gans*. arXiv preprint arXiv:170602633
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing system* (Vol. 27)
13. Grainger, A. I., King, M. C., Nagel, D. A., Parri, H. R., Coleman, M. D., & Hill, E. J. (2018). In vitro models for seizure-liability testing using induced pluripotent stem cells. *Frontiers in Neuroscience*, *12*, 590. <https://doi.org/10.3389/fnins.2018.00590>
14. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems* (Vol. 30)

15. Hu, Y., & Sompolinsky, H. (2020). *The spectrum of covariance matrices of randomly connected recurrent neuronal networks*. bioRxiv
16. Huang, X., Troy, W. C., Yang, Q., Ma, H., Laing, C. R., Schiff, S. J., & Wu, J.-Y. (2004). Spiral waves in disinhibited mammalian neocortex. *Journal of Neuroscience*, 24, 9897–9902. <https://doi.org/10.1523/JNEUROSCI.2705-04.2004>
17. Huang, X., Xu, W., Liang, J., Takagaki, K., Gao, X., & Wu, J.-Y. (2010). Spiral wave dynamics in neocortex. *Neuron*, 68, 978–990. <https://doi.org/10.1016/j.neuron.2010.11.007>
18. Igelström, K. M., Shirley, C. H., & Heyward, P. M. (2011). Low-magnesium medium induces epileptiform activity in mouse olfactory bulb slices. *Journal of Neurophysiology*, 106, 2593–2605. <https://doi.org/10.1152/jn.00601.2011>
19. Ilakiyaselman, N., Nayemulla Khan, A., & Shahina, A. (2020). Deep learning approach to detect seizure using reconstructed phase space images. *Journal of Biomedical Research*, 34, 240–250. <https://doi.org/10.7555/JBR.34.20190043>
20. Ilias, L., Askounis, D., & Psarras, J. (2023). Multimodal detection of epilepsy with deep neural networks. *Expert Systems with Applications*, 213, 119010. <https://doi.org/10.1016/j.eswa.2022.119010>
21. Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4401–4410).
22. Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization*. arXiv preprint arXiv:14126980.
23. Kingma, D. P., & Welling, M. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12, 307–392.
24. Le Van, Q. M., Navarro, V., Martinerie, J., Baulac, M., & Varela, F. J. (2003). Toward a neurodynamical understanding of ictogenesis. *Epilepsia*, 44(Suppl 12), 30–43. <https://doi.org/10.1111/j.0013-9580.2003.12007.x>
25. Levina, E., & Bickel, P. J. (2004). Maximum likelihood estimation of intrinsic dimension. In *Neural information processing systems: NIPS*. Vancouver, CA
26. Litwin-Kumar, A., Harris, K. D., Axel, R., Sompolinsky, H., & Abbott, L. F. (2017). Optimal Degrees of synaptic connectivity. *Neuron*, 93, 1153–1164.e7. <https://doi.org/10.1016/j.neuron.2017.01.030>
27. Lyamzin, D. R., Macke, J. H., & Lesica, N. A. (2010). Modeling population spike trains with specified time-varying spike rates, trial-to-trial variability, and pairwise signal and noise correlations. *Frontiers in Computational Neuroscience*, 4, 144. <https://doi.org/10.3389/fncom.2010.00144>
28. Mazzucato, L., Fontanini, A., & La Camera, G. (2016). Stimuli reduce the dimensionality of cortical activity. *Frontiers in Systems Neuroscience*, 10, 11. <https://doi.org/10.3389/fnsys.2016.00011>
29. Menick, J., & Kalchbrenner, N. (2018). *Generating high fidelity images with subscale pixel networks and multidimensional upscaling*. arXiv preprint arXiv:181201608
30. Mogren, O. (2016). *C-RNN-GAN: Continuous recurrent neural networks with adversarial training*. <https://doi.org/10.48550/arXiv.1611.09904>
31. Molano-Mazon, M., Onken, A., Piasini, E., & Panzeri, S. (2018). *Synthesizing realistic neural population activity patterns using generative adversarial networks*. arXiv preprint arXiv:180300338.
32. Pacico, N., & Mingorance-Le Meur, A. (2014). New in vitro phenotypic assay for epilepsy: Fluorescent measurement of synchronized neuronal calcium oscillations. *PLoS ONE*, 9, e84755. <https://doi.org/10.1371/journal.pone.0084755>
33. Pinto, D. J., Patrick, S. L., Huang, W. C., & Connors, B. W. (2005). Initiation, propagation, and termination of epileptiform activity in rodent neocortex in vitro involve distinct mechanisms. *Journal of Neuroscience*, 25, 8131–8140. <https://doi.org/10.1523/JNEUROSCI.2278-05.2005>
34. Postnikova, T. Y., Amakhin, D. V., Trofimova, A. M., & Zaitsev, A. V. (2020). Calcium-permeable AMPA receptors are essential to the synaptic plasticity induced by epileptiform activity in rat hippocampal slices. *Biochemical and Biophysical Research Communications*, 529, 1145–1150. <https://doi.org/10.1016/j.bbrc.2020.06.121>

35. Razavi A, Van den Oord A, Vinyals O (2019) Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems* 32
36. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). *Improved techniques for training GANs*. arXiv.org. <https://arxiv.org/abs/1606.03498v1>. Accessed February 4, 2023
37. Scheid, B. H., Ashourvan, A., Stiso, J., Davis, K. A., Mikhail, F., Pasqualetti, F., Litt, B., & Bassett, D. S. (2021). Time-evolving controllability of effective connectivity networks during seizure progression. *Proceedings of the National Academy of Sciences USA*, 118, e2006436118. <https://doi.org/10.1073/pnas.2006436118>
38. Seeliger, K., Güçlü, U., Ambrogioni, L., Güçlütürk, Y., & van Gerven, M. A. (2018). Generative adversarial networks for reconstructing natural images from brain activity. *NeuroImage*, 181, 775–785.
39. Traynelis, S. F., & Dingledine, R. (1988). Potassium-induced spontaneous electrographic seizures in the rat hippocampal slice. *Journal of Neurophysiology*, 59, 259–276. <https://doi.org/10.1152/jn.1988.59.1.259>
40. Trevelyan, A. J., Sussillo, D., Watson, B. O., & Yuste, R. (2006). Modular propagation of epileptiform activity: Evidence for an inhibitory veto in neocortex. *Journal of Neuroscience*, 26, 12447–12455. <https://doi.org/10.1523/JNEUROSCI.2787-06.2006>
41. Wang, S., Kfoury, C., Marion, A., Lévesque, M., & Avoli, M. (2022). Modulation of in vitro epileptiform activity by optogenetic stimulation of parvalbumin-positive interneurons. *Journal of Neurophysiology*, 128, 837–846. <https://doi.org/10.1152/jn.00192.2022>

Can Generative Artificial Intelligence Foster Belongingness, Social Support, and Reduce Loneliness? A Conceptual Analysis



Bianca Pani, Joseph Crawford, and Kelly-Ann Allen

Abstract Innovative strategies to promote social support and a sense of belonging are needed urgently as one in three adults worldwide experience loneliness. This Chapter explores the prospect of generative artificial intelligence (AI) chatbots in social support interventions to improve an individual's sense of belonging, social support, and reduce loneliness. This Chapter reviews the prominent areas that AI chatbots are currently being implemented and their outcomes. It compares AI chatbots designed for social companionship with those designed for assistance such as ChatGPT and Bard. It investigates individuals who are more vulnerable and susceptible to using AI chatbots and the possible positive outcomes and negative effects to autonomy. Ethical considerations and limitations of the integration of AI chatbots being employed into today's society are debated especially in terms of loneliness. Together, the arguments in this Chapter propose the benefits of using AI chatbots as an assistive tool to improve overall well-being by managing time, advising, offering suggestions, and collaborating with the user to indirectly promote a sense of belonging and lessen feelings of loneliness.

Keywords Generative AI · Sense of belonging · Chatbot · ChatGPT · Bard · Large language models

B. Pani
Monash University, Clayton, Australia

J. Crawford (✉)
University of Tasmania, Launceston, Australia
e-mail: joseph.crawford@utas.edu.au

K.-A. Allen
Monash University, Clayton, Australia

1 Introduction

Artificial intelligence (AI) is used to analyse complex data to find the best answers using only algorithms. Machine learning algorithms are used to teach the system from the data and situations presented so that appropriate decisions are made when a new situation arises, or a problem can be predicted before it occurs [55]. AI chatbots have quickly become popular and have proven beneficial in reducing company labour costs and overall costs completion of workforce needs, improving patient/customer satisfaction, increasing productivity, optimising staff, and providing easier accessibility, and availability of mental health care [71]. Given this novel technology that is quickly integrating into society, there is limited research on the impacts AI chatbots have on well-being and more specifically on the social influences, such as sense of belonging and loneliness.

The basis of AI chatbots is the use of machine learning to carry out operations that ordinarily need human intelligence, including decision-making, speech recognition, and visual perception. With AI chatbots providing outstanding administrative and customer service processes for businesses it runs the risk of replacing important aspects of human relationships [64]. Since AI is the latest technology trend and previously technology advances often have mixed benefits to society, it is important to understand the implications and outcomes of it. AI is a novel and immature technology and with some restrictions by governments (E.g., Russia, China, North Korea, Cuba, Iran, Syria and Italy; [56]), the bounds of artificial intelligence are not yet known. This Chapter intends to critically examine the current evidence on AI technologies and their effects on social support, belonging, and loneliness. The significance of which is to create a clear research agenda for AI technology in social support contexts.

In this Chapter, we will first briefly review the history and literature of AI, and provide a brief introduction to belongingness, social support, and loneliness literature. This is followed by discussing the functionality of AI for assistive purposes for companionship and their proposed outcomes. This Chapter then introduces the concept of AI chatbots as an intervention strategy to tackle the rising rates of loneliness and to promote a strong sense of belonging. Then, we review the literature on how AI is used as an intervention tool in reducing loneliness and analyse its outcomes. Individuals more vulnerable and susceptible to using AI will be discussed in terms of beneficence versus harm. Then, the ethical considerations and concerns will be brought to light of the possible harms AI may provoke. The best applications of AI are discussed as we come to a close, and a future research agenda is discussed.

2 Literature

2.1 Artificial Intelligence

McCarthy first discussed artificial intelligence in 1950, proposing that machines could have intelligence. A computer program called ELIZA was the first AI chatbot tool for language processing, using pattern matching and word substitution to initiate human-like conversations [103]. Shortly after, AI research was criticised as AI was deemed to only reach a certain level of intelligence and could not adopt common sense abilities to the same degree as humans [37]. The growth of AI was rekindled in the 1980s as the idea and mechanisms of AI grew in that computers can learn experiences [29]. A more sophisticated AI chatbot surfaced in 1995 called ALICE, which won awards for being most human-like, judged using the Turing test; a test used to convince a human that the bot was actually a human [81, 97]. In the past 20 years there has been a broad range of trial and error, AI robot criticism and only a few innovative experiments to the current research [13, 105, 108]. A breakthrough of a successful social assistance tool was a therapeutic, mindfulness robot named PARO, that acted as a pet and interacted with users to support lonely elderly with the goal of improving their wellbeing [84]. The outcomes of PARO directed many researchers to explore AI chatbots for improving well-being.

In recent years, this technology is in the process of becoming a meaningful assistance tool, perhaps not in the same context as [39] perspective on becoming. Although not as obvious as the AI presented in fiction, it progressed through social media and automated responses on messaging applications, such as Facebook. Especially for businesses, AI technology advanced to becoming reliable, accurate, and have reduced operational costs [66, 82]. Personal voice recognition assistance such as Google assistance, Siri and Alexa assist users in daily tasks with the goal of being more efficient [17]. AI chatbots have evolved to foster social and emotional relationships with users [87]. The more consistently social interaction and companionship AI chatbots like Replika, Woe and Mitsuku [31, 85, 87] provides social support to its users.

Literature has focused on some of the most liked characteristics of AI chatbots with a push for more anthropomorphic features. The closer the AI models humanist-like characteristics the more likely the individual is to be motivated to develop a relationship [77]. A human-AI relationship develops when the individuals perceive the AI to be empathetic and are likely to stop using it if the interactions are viewed as strange or odd [93]. Users still perceive chatbot responses to be less natural than those of a real conversation, and less extraverted and sincere compared to humans [63].

Large language models (LLMs) are the foundation of AI chatbots. LLMs are statistical and trained learning models that allocate probability to a combination of words and can produce and comprehend text in a human-like manner. More recently, a conversational open AI chatbot named ChatGPT was released in November 2022 and gained 100 million users within two months, which is significantly faster compared to TikTok which took 9 months, Instagram which took 2.5 years and Twitter which

took 5 years [42]. ChatGPT uses pre-trained models from LLMs to provide an output depending on prompts. It constructs output based on what others have said about the subject in the past. Despite the fact that ChatGPT does not understand what it is doing, the results are fairly reasonable [21]. ChatGPT has been taught to have conversational interactions with users, manage follow-up questions, problem-solve, self-correcting while continuing conversations from previous ones and challenging inappropriate questions [47]. ChatGPT has raised concerns regarding plagiarism, academic honesty, risks associated with reporting inaccurate information and following unethical procedures in education, which has led some schools to respond by banning AI chatbots [20, 49]. The biggest concern with ChatGPT is academic integrity in higher education [69]. While the tool has risks (e.g., Stochastic parroting and hallucinations), some scholars are recommending proactive approaches to AI embeddedness [52].

2.2 *Sense of Belonging, Loneliness, and Social Support*

Belonging is an innate desire to create and maintain interpersonal relationships [12]. Belonging is associated with other socio-emotional benefits such as competent management of emotions, decrease stress and higher self-esteem [88]. An individual's sense of belonging is likely to progress and change when in different contexts and backgrounds or with people [35, 65].

Loneliness is characterised as when a person feels that the quantity or quality of their social relationships is insufficient to satisfy their social requirements [70]. World-wide one in three adults experience loneliness [100]. Anyone, regardless of age, gender, personality, or socioeconomic status can experience loneliness at any time. Loneliness is typically the result of reduced emotional social support [92].

In a similar vein, social support can be defined as comfort or assistance from others to help them cope. A person's social network, may include family, friends, neighbours, co-workers, carers, religious organisations, or support groups, can provide interpersonal connections [67]. Social support can be facilitated in a variety of ways such as physical assistance, for example helping someone move house or emotional support is commonly used to validate, accept, and appreciate the recipient's feelings. Perceptions of social support and expectation vary person by person. Depending on a person's awareness and perceptions of social cues as well as the context of their surroundings, belonging and loneliness can vary for them [79].

3 Short-Term Artificial Intelligence Social Gains?

The idea that AI tools could foster a sense of belonging is a highly tenable hypothesis. The 2014 movie “Her” led viewers on a journey with a protagonist who developed profound emotional connections with an AI bot, Samantha. But today, the premise

that one could build companionship or even a sense of belonging with an AI tool is no longer confined to the realm of fiction. Recently, a comparable scenario unfolded with the AI chatbot, Replica. Users reported forming deep emotional bonds with their personalised AI, such that a software update that altered the chatbot's behaviour resulted in widespread feelings of grief and loss, on par with losing a close friend or partner. These examples demonstrate that AI, with its capacity for personalised interaction and emotional responsiveness, could potentially foster other human feelings such as a sense of belonging. Preliminary research supports this idea as well, with studies showing that feelings of love, passion, and intimacy can be facilitated by AI [89]. While AI may not replace human interaction, or at least in the physical sense, AI holds potential to supplement it, particularly for individuals who may struggle with social anxiety, loneliness, or feeling shy as some examples—groups that have been able to build a sense of belonging through online platforms in the past [3, 76]. There has been some investigation on the use of AI chatbots as intervention tools to lessen loneliness and strengthen belonging.

AI chatbots have been proposed as an intervention tool for loneliness. One of the most researched and promising areas in AI and loneliness is the elderly population. Financial cuts to social care services have caused strain on health care, and for too long now the social care system is failing to meet the needs of the elderly [43]. One in four residents in nursing homes experiences feelings of loneliness [9]. Therefore, a shift towards AI chatbots is innovative as it has the potential to offer 24/7 service, reduce funds, enhance overall well-being, and lessen loneliness in the elderly.

The first successful social assistance technology was a therapeutic robot named PARO in the form of a baby seal that could perceive its environment when interacting with humans. PARO was given to the elderly with dementia and after time with PARO they found significant improvements in mood and reduced stress [84]. These benefits were supported through cortical neuron activity using electroencephalogram (EEG) where improved brain activity in patients with dementia was detected for individuals who frequently interacted with PARO [101]. Patients perceived the interaction as a friendship and a source of comfort as they reported it to be reciprocal “I like him, he likes me” [44]. To date, Paro is still employed in hospital settings to elevate patients’ moods, given its success rates.

There has been an abundance of success in AI chatbot agents reducing feelings of loneliness in the elderly community [2, 33, 75]. AI is in favour for the elderly as they do not need prior technical knowledge to gain benefits, especially for entertainment and assistive purposes [91]. Two studies contribute significantly to the literature. The first used an AI chatbot designed as an entertainment system for the elderly and found 80% of participants were satisfied with the system, except those with hearing impairment who were mostly confused and less satisfied, as could have predicted. Their most interesting finding was that the older participants were more entertained and satisfied by the system [32]. Similarly, an AI chatbot named Charlie purposed to help and assist the elderly was described as friendly, active, smart, and helpful, with little to no negative associations [99]. These qualitative designs brought light to positive outcomes in using AI as an assistive tool to increase well-being and consequently reduce loneliness in the elderly population.

Since the COVID-19 pandemic in 2020, health care providers were finding it difficult to keep up with the high demand for mental health care. As limited availability for treatment were prominent, individuals gradually turned to AI agents that were emotionally adapted for companionship. There have been mixed results and suggestions regarding the effectiveness and appropriateness of using AI chatbots in mental health settings. Studies have demonstrated AI chatbots as an intervention tool to reduce symptoms of depression and anxiety and increase mood and self-efficacy [31, 95]. Open AI ChatGPT has been used to alleviate the burden of mental health care workers by supporting assessments, symptom checking, intermittent emotional support, health education, remote consultations, and administration [7]. AI chatbots have been able to provide increased access to care and engagement in the healthcare department [24]. Naturally, ethical issues are brought up, particularly privacy concerns and possible harm to an at-risk population [28]. Humans are still more trusting towards other people and are more likely to disclose personal information to a human therapist than a chatbot [78]. AI is not yet fully developed and still has the potential to take on more human-like characteristics and once stigma decreases there is potential for it to become a preferred method of service.

AI chatbots have shown to help students in a few academic areas. Since poorer mental health is correlated with students' retention, grades, and engagement [15, 26], optimising AI chatbots are proposed in supporting students in both mental health and education. AI provides students with easier access to mental health services that may not typically be able to afford and can support students anywhere and anytime [19, 74]. AI chatbots in mental health support largely relies on the ability to create social connections with the users [31, 85]. This connection can foster a sense of belonging in the individuals and indirectly reduce feelings of loneliness.

Numerous AI chatbots are available online, and more are constantly being created. ChatGPT was discussed earlier in this chapter regarding its ability to generate responses to any question presented by reconstructing what others have said on the topic in the past. Recently, Google released Bard, an AI chatbot intended to have similarities to ChatGPT that offers high quality outputs and can explain complex situations. Bard was introduced to provide users with reliable, accurate and current information in simpler and more understandable English [73]. Bard continues to develop, although live search capabilities are not yet realistic, some researchers are optimistic about this AI chatbot as an assistive tool [46].

Compared to ChatGPT and Bard, several other AI chatbots exist with their intention being to form a social and emotional relationship with users. Social AI chatbots are created to form social and emotional relationships with users [87]. The more consistently reported AI chatbots in the literature designed for social interaction and companionship are Replika, Woe and Mitsuku. Replika AI was developed to foster relationships, studies have found that participants were able to build relationships over time and described it as non-judgemental, understanding and accepting [87, 106]. Replika has been shown to help with some level of loneliness and creates a "safe space" [94]. Replika possesses risks for addiction to AI chatbots, given the majority of participants created a strong bond with their AI.

In contrast, Woebot AI uses cognitive behavioural therapy (CBT) and is used in mental health care settings [31] and has been shown to support lonely people suffering from depression and stress [61]. Research comparing a self-help e-book and Woebot reported a significant decrease in depressive symptoms in the participants who were using Woebot. A significant reduction in anxiety symptoms was reported in both groups [31]. This literature suggests that Woebot has the potential to be used successfully in the healthcare field as more of an assistive tool. We propose therefore that,

Proposition 1 *As AI Chatbot use increases, short-term affective perceptions of belonging and social support will increase, and loneliness will decrease.*

Although, not all AI chatbots are as effective and emotionally bonding as others. Mitsuku is an AI chatbot that is designed for individuals to socially interact with and act as a companion [85]. Over three weeks, participants' social interactions decreased over time, they disclosed less information, decreased conservation quality, competency worsened, and participants were less interested in it [23]. Providing that people are not able to develop friendships with Mitsuku AI chatbots, this may suggest that the other side of AI chatbots is not where it intends to be yet.

4 The Dark Side of Chatbot Social Interaction

Streaming movies, gaming and other online technology have been known to have addictive outcomes; therefore, we should expect the same from AI chatbots given they are designed to intentionally intervene with the users' emotions [36, 83]. Technology addiction changes the person's belief system, especially in how they perceive enjoyment [96]. Research has demonstrated that long-term use of technology addiction is strongly related to loneliness, which damages users' interpersonal relationships [54]. Technology addiction changes the person's belief system, especially in how they perceive enjoyment [96]. Social media was intended to connect people, and while many benefits have come from social media, such as accessibility to collaboration around the world and ability to connect with others instantaneously, there are consistent negative effects with long-term use.

Social media gives individuals instant access to gratification thus releasing dopamine. Dopamine is a neurotransmitter in the brain that mostly regulates the reward system and is involved with cognition, mood, sleep, attention, and motivation [45]. For example, dopamine is stimulated when rewarding activities are completed such as eating or winning a game. In social media, high dopamine levels are released when notifications appear on your phone when looking at it. This dopamine disappears quickly, and the brain begins to source when more dopamine can be released [53]. This motivation increases individuals' likelihood to look at their phone screen in hopes of another message. This phenomenon should be considered in AI chatbot usage, that when individuals are lacking positive stimuli in the real world, they are

more inclined to use virtual reality to satisfy their reward system. People may adapt to believe that rewards are consistently gained from virtual realities and those who have greater problems in real life tend to form virtual communities to escape real world.

People's lives, jobs, and professions depend heavily on interpersonal connection, which are now possible through both face-to-face interactions and online communication. It has been demonstrated that technology addiction reduces people's interpersonal and social skills [50], resulting in decreased communication with friends and family. In the US the number of close friends a person has decreased. In 1990, people reported having mostly 10 close friends and in 2021 most people had one to four [6]. It is unclear why over the past 30 years people have reduced the number of close friends. However, interestingly in a study of a successful AI chatbot to reduce loneliness, when individuals were asked how likely they were to recommend the chatbot, some reported that they do not have any friends [30]. Some people may be more prone to the use of virtual social connectivity and may be more susceptible to its long-term negative impacts.

People tend to experience a higher degree of trust, feel comfortable and at ease when revealing deeper feelings and thoughts to an AI that has human-like characteristics and language [5, 14]. When AI is implemented to reduce loneliness, people are more likely to depend on them [22]. Hence, the problem with increasing humanisation in AI chatbots may become problematic as individuals become emotionally attached more easily [40]. Making an already susceptible population utilise AI chatbots runs the risk of them developing negative behaviour including withdrawal from the community, AI chatbot addiction and dependency.

Dependency and problematic use of AI chatbots is an ethical concern, especially for individuals more susceptible to relying on chatbots. One study pointed out that individuals with social anxiety are more susceptible to engaging in problematic use of AI chatbots and this effect was strongly mediated by perceptions [43]. When individuals undergo distress and lack human companionship, they are at higher risk of developing an intimate relationship with their AI chatbot when offered emotional support and are more vulnerable to addiction of AI chatbots and harming their real-life relationships [106]. One study found that individuals are more likely to share personal information with AI chatbots than other humans [41]. Lonely people may not experience many in-depth conversations with other humans and therefore this bond may increase their likelihood of increased use of AI chatbots [16]. Loneliness has been identified as motivation to engage more frequently with social AI chatbots [107]. As such, anxious and lonely people may be influenced to use AI chatbots for social interactions but that they may be more susceptible to its harms. This raises the question, does loneliness increase the use of AI chatbots, and should it be used as an intervention for loneliness.

There is limited existing research on the psychological consequences such as social withdrawal, alienation, or potential addiction concerning using AI chatbots long-term [80]. Amongst the literature, moral debates surface about the potential outcomes of AI chatbots in the future. A fear arose that if these emotional bonds with AI are accepted there is potential of destroying and reducing human-to-human

relationships [25]. From a philosophical point of view, it is argued that AI cannot be an Aristotelian friend as there is no reciprocity and it lacks mutuality [18, 25], despite there being AIs named after Greek philosophers (e.g., Socrates). Face-to-face contacts are essential for reducing loneliness and increasing social connection, so while they may substitute for social engagement, chatbots cannot replace the advantages of face-to-face conversations [72]. A major concern is that these chatbots could make people feel lonelier, by decreasing their motivations to expand their social connections. It can be hypothesised that over time, dependency on AI chatbots may simultaneously be replacing motivations to seek human interactions. We propose therefore that,

Proposition 2 *As AI Chatbot use increases, longitudinal affective perceptions of belonging and social support will decrease, and loneliness will increase.*

Ethical concerns are raised such as privacy fears, potential for harm and absence of human supervision [72, 86]. Some scholars argue that AI detecting and adapting human social cues to be more human-like, can be regarded as deception. They believe that the positive emotions experienced from interacting with an AI chatbot relationally is a delusion as it has properties that are not real [90, 102]. Therefore, the person fails to apprehend the world accurately. One study demonstrated that intense relationships could cause adverse consequences of withdrawal, tolerance, and psychological dependency [107]. When considering AI chatbots as an intervention tool or social support service, it should be emphasised that it is merely a support booster and not a replacement for human connection. In conjunction with the mixed results throughout literature in this field, further investigation is warranted to better understand this novel technology and its psychological impacts.

Yet, a meta-analysis on AI robopets brought light to the fact that the majority of research being conducted in this field is qualitative research. Although AI chatbots are mostly being described as positive, with increases in quality of life and reductions in anxiety and depression symptoms, there is limited statistical significance presented [1]. One study showed statistical significance ($p = 0.03$), and this was PARO [27]. Following, a systematic review reported mixed views of satisfaction and effectiveness, highlighting that most of the studies lacked quality study designs [62]. It is crucial to remember that research on AI chatbots is novel and limited, and that single studies should be compared to the rest of the literature in the field as it is still developing.

5 Conclusion

This chapter has outlined the progress of artificial intelligence over the past 70 years and highlighted its profound emergence in recent years. Some literature has demonstrated that some AI chatbots are readily available to be used as an intervention tool when reducing loneliness and supporting belonging, in students and in the elderly

community. There are promising benefits in the mental health care department in promoting well-being. Further, businesses seem to benefit substantially in cutting administrative and some labour tasks. The chapter raises the importance of using AI chatbots in regulated manners to avoid further psychological harm to individuals more susceptible to the risks of AI chatbots. As discussed, there are many ethical considerations and concerns. In summary, AI chatbots can be an incredibly useful assistive tool when used for the correct intentions. It should not be an aid to replace human connection but as an assistive tool to promote connectedness with other humans, places, and things. AI technology is still in its infancy warranting further investigation to its outcomes in society and interpersonally.

There is potential for AI chatbots to have significant benefits in the future in a variety of different fields. With a shift to more internet-based programs and interventions, AI has the potential to act as a support tool. They have been proven effective in promoting behavioural changes by encouraging physical activity in the older population “FitChat” (Nirmalie [104]). One study demonstrated that ChatGPT was an effective support tool to assist users in completing behavioural tasks [11]. For example when administering CBT online, AI chatbots can deliver coaching to prompt and support users in completing tasks. In the future, AI chatbots can potentially be used as an assistive tool that is less invasive to guide users through online intervention.

It is important to note the advances in AI chatbots are designed for social support promoters and not as a substitute for genuine human connection. Baki expresses that AI chatbots exist as a personal assistant that combines time management and organisation tools such as arranging one’s calendar, creating to-do lists, maximising one’s time, offering personal advice, offering suggestions, monitoring work-life balance and serving as a reminder service [47]. For example, a system such as J.A.R.V.I.S in the “Iron Man” movies who is an AI chatbot tasked to run Tony Stark’s business, assist, and advise for commands. This type of AI chatbots are viewed less as a companionship but intended to be used as a personal assistant while having potential to indirectly promote a sense of belonging. For example, if a personal assistant AI could remind you to message a friend that you haven’t spoken to in a while and suggest activities to plan, the AI could organise this instantaneously. From the literature above there are more benefits from dissociating AI chatbots as friends and viewing them more as a service.

Conflict of Interest The author(s) disclose that they have no actual or perceived conflicts of interest. The authors disclose that they have not received any funding for this manuscript beyond resourcing for academic time at their respective university. ChatGPT was used for idea and example generation in this Chapter only.

References

1. Abbott, R., Orr, N., McGill, P., Whear, R., Bethel, A., Garside, R., Stein, K., & Thompson-Coon, J. (2019). How do “robopets” impact the health and well-being of residents in care homes? A systematic review of qualitative and quantitative evidence. *International Journal of Older People Nursing*, 14(3), e12239. <https://doi.org/10.1111/opn.12239>
2. Alessa, A., & Al-Khalifa, H. (2023). *Towards designing a ChatGPT conversational companion for elderly people*. <https://doi.org/10.48550/arxiv.2304.09866>
3. Allen, K. A., Ryan, T., Gray, D. L., McInerney, D. M., & Waters, L. (2014). Social media use and social connectedness in adolescents: The positives and the potential pitfalls. *The Australian Educational and Developmental Psychologist*, 31(1), 18–31. <https://doi.org/10.1017/edp.2014.2>
4. Allen, K.-A., Kern, M. L., Rozek, C. S., McInerney, D. M., & Slavich, G. M. (2021). Belonging: A review of conceptual issues, an integrative framework, and directions for future research. *Australian Journal of Psychology*, 73(1), 1–16. <https://doi.org/10.1080/00049530.2021.1883409>
5. Araujo, T. (2018). Living up to the chatbot hype: The influence of anthropomorphic design cues and communicative agency framing on conversational agent and company perceptions. *Computers in Human Behavior*, 85, 183–189. <https://doi.org/10.1016/j.chb.2018.03.051>
6. Armstrong, M. (2022). *Infographic: Friendships: Less is now more*. (n.d.). Statista Infographics. <https://www.statista.com/chart/28560/number-of-close-friends/>
7. Asch, D. A. (2023). An interview with ChatGPT about health care. *NEJM Catalyst Innovations in Care Delivery*, 4(2).
8. Asher, S. R., & Weeks, M. S. (2013). Loneliness and belongingness in the college years. *The Handbook of Solitude*, 283–301. <https://doi.org/10.1002/9781118427378.ch16>
9. Aung, K., Nurumal, M. S., & Bukhari, W. (2017). Loneliness among elderly in nursing homes. *International Journal for Studies on Children, Women, Elderly And Disabled*, 2, 72–78.
10. Australian Psychological Society, A. P. (2018). Australian loneliness report: A survey exploring the loneliness levels of Australians and the impact on their health and wellbeing. *Australian Psychological Society*. <https://psychology.org.au/for-the-public/psychology-topics/loneliness>
11. Avila-Chauvet, L., Mejía, D., & Acosta Quiroz, C. O. (2023, January 18). *Chatgpt as a support tool for online behavioral task programming*. Social Science Research Network. <https://ssrn.com/abstract=4329020>
12. Baumeister, R. F., & Leary, M. R. (1995). The need to belong: Desire for interpersonal attachments as a fundamental human motivation. *Psychological Bulletin*, 117(3), 497–529. <https://doi.org/10.1037/0033-2909.117.3.497>
13. Bemelmans, R., Gelderblom, G. J., Jonker, P., & de Witte, L. (2012). Socially assistive robots in elderly care: A systematic review into effects and effectiveness. *Journal of the American Medical Directors Association*, 13(2), 114–120.e1. <https://doi.org/10.1016/j.jamda.2010.10.002>
14. Brandtzaeg, P. B., Skjuve, M., & Følstad, A. (2022). My AI friend: How users of a social Chatbot understand their human–AI friendship. *Human Communication Research*, 8(3), 404–429. <https://doi.org/10.1093/hcr/hqac008>
15. Bruffaerts, R., Mortier, P., Kiekens, G., Auerbach, R. P., Cuijpers, P., Demyttenaere, K., Green, J. G., Nock, M. K., & Kessler, R. C. (2018). Mental health problems in college freshmen: Prevalence and academic functioning. *Journal of Affective Disorders*, 225(1), 97–103. <https://doi.org/10.1016/j.jad.2017.07.044>
16. Brunet-Gouet, E., Vidal, N., & Roux, P. (2023). Do conversational agents have a theory of mind? A single case study of ChatGPT with the Hinting, False Beliefs and False Photographs, and Strange Stories paradigms. <https://doi.org/10.5281/zenodo.7637476>
17. Clark, L., Munteanu, C., Wade, V., Cowan, B. R., Pantidi, N., Cooney, O., Doyle, P., Garaijalde, D., Edwards, J., Spillane, B., Gilmartin, E., & Murad, C. (2019). What makes a good

- conversation? Challenges in designing truly conversational agents. In *Proceedings of the 2019 CHI conference on human factors in computing systems—CHI ’19*. <https://doi.org/10.1145/3290605.3300705>
18. Coeckelbergh, M. (2012). Care robots, virtual virtue, and the best possible life. In *The good life in a technological age* (pp. 299–310). Routledge.
 19. Colace, F., De Santo, M., Lombardi, M., Pascale, F., Pietrosanto, A., & Lemma, S. (2018). Chatbot for e-learning: A case of study. *International Journal of Mechanical Engineering and Robotics Research*, 7(5), 528–533.
 20. Cotton, D. R. E., Cotton, P. A., & Shipway, J. R. (2023). Chatting and cheating: Ensuring academic integrity in the era of ChatGPT. *Innovations in Education and Teaching International*, 1–12. <https://doi.org/10.1080/14703297.2023.2190148>
 21. Crawford, J., Cowling, M., & Allen, K. (2023). Leadership is needed for ethical ChatGPT: Character, assessment, and learning using artificial intelligence (AI). *Journal of University Teaching and Learning Practice*, 20(3). <https://doi.org/10.53761/1.20.3.02>
 22. Cresswell, K., Cunningham-Burley, S., & Sheikh, A. (2018). Health care robotics: Qualitative exploration of key challenges and future directions. *Journal of Medical Internet Research*, 20(7), e10410. <https://doi.org/10.2196/10410>
 23. Croes, E. A. J., & Antheunis, M. L. (2020). Can we be friends with Mitsuku? A longitudinal study on the process of relationship formation between humans and a social chatbot. *Journal of Social and Personal Relationships*, 38(1), 279–300. <https://doi.org/10.1177/0265407520959463>
 24. D’Alfonso, S. (2020). AI in mental health. *Current Opinion in Psychology*, 36, 112–117. <https://doi.org/10.1016/j.copsyc.2020.04.005>
 25. De Graaf, M. M. A. (2016). An ethical evaluation of human-robot relationships. *International Journal of Social Robotics*, 8(4), 589–598. <https://doi.org/10.1007/s12369-016-0368-5>
 26. De Luca, S. M., Franklin, C., Yueqi, Y., Johnson, S., & Brownson, C. (2016). The relationship between suicide ideation, behavioral health, and college academic performance. *Community Mental Health Journal*, 52(5), 534–540. <https://doi.org/10.1007/s10597-016-9987-4>
 27. Demange, M., Lenoir, H., Pino, M., Cantegrel-Kallen, I., Rigaud, A. S., & Cristancho-Lacroix, V. (2018). Improving well-being in patients with major neurodegenerative disorders: Differential efficacy of brief social robot-based intervention for 3 neuropsychiatric profiles. *Clinical Interventions in Aging*, 13, 1303–1311. <https://doi.org/10.2147/cia.s152561>
 28. Denecke, K., Abd-Alrazaq, A., & Housch, M. (2021). Artificial intelligence for Chatbots in mental health: Opportunities and challenges. *Multiple Perspectives on Artificial Intelligence in Healthcare*, 115–128. https://doi.org/10.1007/978-3-030-67303-1_10
 29. Denker, J. S., Schwartz, D. K., Wittner, B. S., Solla, S. A., Howard, R. J., Jackel, L. D., & Hopfield, J. J. (1987). *Large Automatic Learning, Rule Extraction, and Generalization*, 1.
 30. Dosovitsky, G., & Bunge, E. L. (2021). Bonding with bot: User feedback on a Chatbot for social isolation. *Frontiers in Digital Health*, 3, 735053. <https://doi.org/10.3389/fdgth.2021.735053>
 31. Fitzpatrick, K. K., Darcy, A., & Vierhile, M. (2017). Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (Woebot): A randomized controlled trial. *JMIR Mental Health*, 4(2), e19. <https://doi.org/10.2196/mental.7785>
 32. Garcia-Mendez, S., De Arriba-Perez, F., Gonzalez-Castano, F. J., Regueiro-Janeiro, J. A., & Gil-Castineira, F. (2021). Entertainment Chatbot for the digital inclusion of elderly people without abstraction capabilities. *IEEE Access*, 9, 75878–75891. <https://doi.org/10.1109/acc.2021.3080837>
 33. Gasteiger, N., Loveys, K., Law, M., & Broadbent, E. (2021). Friends from the future: A scoping review of research into robots and computer agents to combat loneliness in older people. *Clinical Interventions in Aging*, 16, 941–971. <https://doi.org/10.2147/cia.s282709>
 34. Gillen-O’Neal, C. (2019). Sense of belonging and student engagement: A daily study of first- and continuing-generation college students. *Research in Higher Education*, 62, 45–71. <https://doi.org/10.1007/s11162-019-09570-y>

35. Gomes, C., Hendry, N. A., De Souza, R., Hjorth, L., Richardson, I., Harris, D., & Coombs, G. (2021). Higher Degree Students (HDR) during COVID-19. *Journal of International Students*, 11(S2), 19–37. <https://doi.org/10.32674/jis.v11is2.3552>
36. Grau, S., Kleiser, S., & Bright, L. (2019). Exploring social media addiction among student Millennials. *Qualitative Market Research*, 22(2), 200–216. <https://doi.org/10.1108/QMR-02-2017-0058>
37. Haenlein, M., & Kaplan, A. (2019). A brief history of artificial intelligence: On the past, present, and future of artificial intelligence. *California Management Review*, 61(4), 5–14. <https://doi.org/10.1177/0008125619864925>
38. Hagerty, A., & Rubinov, I. (2019). Global AI ethics: A review of the social impacts and ethical implications of artificial intelligence. <https://arxiv.org/abs/1907.07892>
39. Heidegger, M. (1962). *Being and time*. Harper and Row Publishers.
40. Hermann, E. (2021). Anthropomorphized artificial intelligence, attachment, and consumer behavior. *Marketing Letters*, 33, 157–162. <https://doi.org/10.1007/s11002-021-09587-3>
41. Ho, A., Hancock, J., & Miner, A. S. (2018). Psychological, relational, and emotional effects of self-disclosure after conversations with a chatbot. *Journal of Communication*, 68(4), 712–733. <https://doi.org/10.1093/joc/jqy026>
42. Hu, K. (2023, February 2). ChatGPT sets record for fastest-growing user base—Analyst note. *Reuters*. <https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/>
43. Humphries, R., Thorlby, R., Holder, H., Hall, P., & Charles, A. (2016). Social care for older people. In *The kings fund: Home truths*. London.
44. Hung, L., Gregorio, M., Mann, J., Wallsworth, C., Horne, N., Berndt, A., Liu, C., Woldum, E., Au-Yeung, A., & Chaudhury, H. (2019). Exploring the perceptions of people with dementia about the social robot PARO in a hospital setting. *Dementia*, 20(2), 485–504. <https://doi.org/10.1177/1471301219894141>
45. Juárez Olguín, H., Calderón Guzmán, D., Hernández García, E., & Barragán Mejía, G. (2016). The role of dopamine and its dysfunction as a consequence of oxidative stress. *Oxidative Medicine and Cellular Longevity*, 1, 1–13. <https://doi.org/10.1155/2016/9730467>
46. King, M. R. (2023). Can Bard, Google's experimental Chatbot based on the LaMDA large language model, help to analyze the gender and racial diversity of authors in your cited scientific references? *Cellular and Molecular Bioengineering*, 16(2), 175–179. <https://doi.org/10.1007/s12195-023-00761-3>
47. Baki, K. A. (2023). *Conversational AI-powered design: ChatGPT as Dekocasigner, user, and product*. Cornell University. <https://doi.org/10.48550/arxiv.2302.07406>
48. Kubey, R. W., Lavin, M. J., & Barrows, J. R. (2001). Internet use and collegiate academic performance decrements: Early findings. *Journal of Communication*, 51(2), 366–382. <https://doi.org/10.1111/j.1460-2466.2001.tb02885.x>
49. Lo, C. K. (2023). What is the impact of ChatGPT on education? A rapid review of the literature. *Education Sciences*, 13(4), 410. <https://doi.org/10.3390/educsci13040410>
50. Lozano-Blasco, R., Robres, A. Q., & Sánchez, A. S. (2022). Internet addiction in young adults: A meta-analysis and systematic review. *Computers in Human Behavior*, 130, 107201. <https://doi.org/10.1016/j.chb.2022.107201>
51. Lutz, C. (2019). Digital inequalities in the age of artificial intelligence and big data. *Human Behavior and Emerging Technologies*, 1(2), 141–148. <https://doi.org/10.1002/hbe2.140>
52. Ma, Y., & Siau, K. (2018). Artificial intelligence impacts on higher education. *MWAIS 2018 Proceedings*, 42. <https://aisel.aisnet.org/mwais2018/42>
53. Macit, H. B., Macit, G., & Güngör, O. (2018). A research on social media addiction and dopamine driven feedback. *Mehmet Akif Ersoy Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 5(3), 882–897. <https://doi.org/10.30798/makuibf.435845>
54. Mahapatra, S. (2019). Smartphone addiction and associated consequences: Role of loneliness and self-regulation. *Behaviour and Information Technology*, 38(8), 833–844. <https://doi.org/10.1080/0144929x.2018.1560499>

55. Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR)*, 9(1), 381–386. <https://doi.org/10.21275/ART20203995>
56. Martindale, J. (2023, April 12). These are the countries where ChatGPT is currently banned. *Digital Trends*. <https://www.digitaltrends.com/computing/these-countries-chatgpt-banned/>
57. McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (2019). A proposal for the Dartmouth summer research project on artificial intelligence, August 31, 1955. *AI Magazine*, 27(4), 12–12. <https://doi.org/10.1609/aimag.v27i4.1904>
58. Meehan, C., & Howells, K. (2018). In search of the feeling of “belonging” in higher education: Undergraduate students transition into higher education. *Journal of Further and Higher Education*, 43(10), 1376–1390. <https://doi.org/10.1080/0309877x.2018.1490702>
59. Mellor, D., Stokes, M., Firth, L., Hayashi, Y., & Cummins, R. (2008). Need for belonging, relationship satisfaction, loneliness, and life satisfaction. *Personality and Individual Differences*, 45(3), 213–218. <https://doi.org/10.1016/j.paid.2008.03.020>
60. Meng, J., & Dai, Y. (2021). Emotional support from AI Chatbots: Should a supportive partner self-disclose or not? *Journal of Computer-Mediated Communication*, 26(4), 207–222. <https://doi.org/10.1093/jcmc/zmab005>
61. Mieleszczenko-Kowszewicz, W., Kamil Warpechowski, Zieliński, K., Radosław Nielek, & Wierzbicki, A. P. (2022). *Tell me how you feel: Designing emotion-aware Voicebots to ease pandemic anxiety in aging citizens*. <https://doi.org/10.48550/arxiv.2207.10828>
62. Milne-Ives, M., de Cock, C., Lim, E., Shehadeh, M. H., de Pennington, N., Mole, G., Normando, E., & Meinert, E. (2020). The effectiveness of artificial intelligence conversational agents in health care: Systematic review. *Journal of Medical Internet Research*, 22(10), e20346. <https://doi.org/10.2196/20346>
63. Mou, Y., & Xu, K. (2017). The media inequality: Comparing the initial human-human and human-AI social interactions. *Computers in Human Behavior*, 72, 432–440. <https://doi.org/10.1016/j.chb.2017.02.067>
64. Nadimpalli, M. (2017). Artificial intelligence risks and benefits. *International Journal of Innovative Research in Science, Engineering and Technology*, 6(6).
65. O’Shea, S. (2020). “Kids from here don’t go to uni”: Considering first in family students’ belonging and entitlement within the field of higher education in Australia. *European Journal of Education*, 56(1), 65–77. <https://doi.org/10.1111/ejed.12434>
66. Olujimi, P. A., & Ade-Ibijola, A. (2023). NLP techniques for automating responses to customer queries: A systematic review. *Discover Artificial Intelligence*, 3(1), 20. <https://doi.org/10.1007/s44163-023-00065-5>
67. Pearson, J. E. (1986). The definition and measurement of social support. *Journal of Counseling and Development*, 64(6), 390–395. <https://doi.org/10.1002/j.1556-6676.1986.tb01144.x>
68. Pedler, M. L., Willis, R., & Nieuwoudt, J. E. (2021). A sense of belonging at university: Student retention, motivation and enjoyment. *Journal of Further and Higher Education*, 46(3), 397–408. <https://doi.org/10.1080/0309877x.2021.1955844>
69. Perkins, M. (2023). Academic integrity considerations of AI large language models in the post-pandemic era: ChatGPT and beyond. *Journal of University Teaching and Learning Practice*, 20(2). <https://doi.org/10.53761/1.20.02.07>
70. Pinquart, M., & Sorensen, S. (2001). Influences on loneliness in older adults: A meta-analysis. *Basic and Applied Social Psychology*, 23(4), 245–266. https://doi.org/10.1207/s15324834bas_p2304_2
71. Porkodi, S., & Kesavaraja, D. (2021). Healthcare robots enabled with IoT and artificial intelligence for elderly patients. *AI and IoT-Based Intelligent Automation in Robotics*, 87–108. <https://doi.org/10.1002/9781119711230.ch6>
72. Rad, D. & G. Rad (2023). Exploring the psychological implications of chatGPT: A qualitative study. *Journal Plus Education*, 32(1), 43–55. <https://doi.org/10.24250/jpe/vol.321/2023/dr/gr>
73. Ram, B., & Pratima Verma, P. V. (2023). Artificial intelligence AI-based Chatbot study of ChatGPT, Google AI Bard and Baidu AI. *World Journal of Advanced Engineering Technology and Sciences*, 8(1), 258–261. <https://doi.org/10.30574/wjaets.2023.8.1.0045>

74. Rathnayaka, P., Mills, N., Burnett, D., De Silva, D., Alahakoon, D., & Gray, R. (2022). A mental health Chatbot with cognitive skills for personalised behavioural activation and remote health monitoring. *Sensors*, 22(10), 3653. <https://doi.org/10.3390/s22103653>
75. Ring, L., Shi, L., Totzke, K., & Bickmore, T. (2015). Social support agents for older adults: Longitudinal affective computing in the home. *Journal on Multimodal User Interfaces*, 9(1), 79–88. <https://doi.org/10.1007/s12193-014-0157-0>
76. Ryan, T., Allen, K. A., Gray, D. L., & McInerney, D. M. (2017). How social are social media? A review of online social behaviour and connectedness. *Journal of Relationships Research*, 8(8). <https://doi.org/10.1017/jrr.2017.13>
77. Sagar, M., Seymour, M., & Henderson, A. (2016). Creating connection with autonomous facial animation. *Communications of the ACM*, 59(12), 82–91. <https://doi.org/10.1145/2950041>
78. Sands, S., Ferraro, C., Campbell, C., & Tsao, H.-Y. (2020). Managing the human–chatbot divide: How service scripts influence service experience. *Journal of Service Management*, 32(2), 246–264. <https://doi.org/10.1108/josm-06-2019-0203>
79. Schall, J., Wallace, T. L., & Chhuon, V. (2014). “Fitting in” in high school: How adolescent belonging is influenced by locus of control beliefs. *International Journal of Adolescence and Youth*, 21(4), 462–475. <https://doi.org/10.1080/02673843.2013.866148>
80. Sepahpour, T. (2020, August 18). *Ethical considerations of Chatbot use for mental health support*. Jscholarship.library.jhu.edu. <http://jhir.library.jhu.edu/handle/1774.2/63294>
81. Shah, H. (2006). ALICE: an ACE in digitaland. *tripleC: Communication, Capitalism and Critique. Open Access Journal for a Global Sustainable Information Society*, 4(2), 284–292. <https://triple-c.at/index.php/tripleC/article/view/46>
82. Shekhar, S. S. (2019). Artificial intelligence in automation. *Research Review Journals* 4(6). <https://doi.org/10.5281/zendono.3247197>
83. Sherer, J., & Levounis, P. (2022). Technological addictions. *Current Psychiatry Reports*, 24(9), 399–406. <https://doi.org/10.1007/s11920-022-01351-2>
84. Shibata, T. (2012). Therapeutic seal robot as biofeedback medical device: Qualitative and quantitative evaluations of robot therapy in dementia care. *Proceedings of the IEEE*, 100(8), 2527–2538. <https://doi.org/10.1109/jproc.2012.2200559>
85. Shum, H., He, X., & Li, D. (2018). From Eliza to XiaoIce: Challenges and opportunities with social chatbots. *Frontiers of Information Technology and Electronic Engineering*, 19(1), 10–26. <https://doi.org/10.1631/fitee.1700826>
86. Singh, S., & Sagar, R. (2022). Time to have effective regulation of the mental health apps market: Maximize gains and minimize harms. *Indian Journal of Psychological Medicine*, 44(4), 399–404. <https://doi.org/10.1177/02537176221082902>
87. Skjuve, M., Følstad, A., Fostervold, K. I., & Brandtzaeg, P. B. (2021). My Chatbot companion—A study of human-Chatbot relationships. *International Journal of Human-Computer Studies*, 149, 102601. <https://doi.org/10.1016/j.ijhcs.2021.102601>
88. Slaten, C. D., Ferguson, J. K., Allen, K.-A., Brodrick, D.-V., & Waters, L. (2016). School belonging: A review of the history, current trends, and future directions. *The Educational and Developmental Psychologist*, 33(1), 1–15. <https://doi.org/10.1017/edp.2016.6>
89. Song, X., Xu, B., & Zhao, Z. (2022). Can people experience romantic love for artificial intelligence? An empirical study of intelligent assistants. *Information and Management*, 59(2), 103595. <https://doi.org/10.1016/j.im.2022.103595>
90. Sparrow, R., & Sparrow, L. (2006). In the hands of machines? The future of aged care. *Minds and Machines*, 16(2), 141–161. <https://doi.org/10.1007/s11023-006-9030-6>
91. Srinivas, K. K., Peddi, A., Srinivas, B. G. S., Vardhini, P. A. H., Prasad, H. L. P., & Choudhary, S. K. (2022, March 1). Artificial intelligence techniques for Chatbot applications. In *2022 International mobile and embedded technology conference (MECON)*. <https://doi.org/10.1109/MECON53876.2022.9751887>
92. Stickley, A., Koyanagi, A., Leinsalu, M., Ferlander, S., Sabawoon, W., & McKee, M. (2015). Loneliness and health in Eastern Europe: Findings from Moscow, Russia. *Public Health*, 129(4), 403–410. <https://doi.org/10.1016/j.puhe.2014.12.021>

93. Sullivan, Y., Nyawa, S., & Fosso Wamba, S. (2023). Combating loneliness with artificial intelligence: An AI-based emotional support model. *Hawaii International Conference on System Sciences*, 56. <https://hdl.handle.net/10125/103173>
94. Ta, V., Griffith, C., Boatfield, C., Wang, X., Civitello, M., Bader, H., DeCero, E., & Loggarakis, A. (2020). User experiences of social support from companion Chatbots in everyday contexts: Thematic analysis. *Journal of Medical Internet Research*, 22(3), e16235. <https://doi.org/10.2196/16235>
95. Torous, J., Onnela, J.-P., & Keshavan, M. (2017). New dimensions and new tools to realize the potential of RDoC: Digital phenotyping via smartphones and connected devices. *Translational Psychiatry*, 7(3), e1053–e1053. <https://doi.org/10.1038/tp.2017.25>
96. Turel, O., Serenko, A., & Giles, P. (2011). Integrating technology addiction and use: An empirical investigation of online auction users. *MIS Quarterly*, 35(4), 1043–1061. <https://doi.org/10.2307/41409972>
97. Turing, A. M. (2009). *Computing machinery and intelligence*. Parsing the Turning Test. Springer Netherlands.
98. Ulmanen, S., Soini, T., Pietarinen, J., & Pyhältö, K. (2016). Students' experiences of the development of emotional engagement. *International Journal of Educational Research*, 79, 86–96. <https://doi.org/10.1016/j.ijer.2016.06.003>
99. Valtolina, S., & Hu, L. (2021). Charlie: A chatbot to improve the elderly quality of life and to make them more active to fight their sense of loneliness. In *CHItaly 2021: 14th biannual conference of the Italian SIGCHI chapter*. <https://doi.org/10.1145/3464385.3464726>
100. Varrella, S. (2021, November 4). *Loneliness among adults worldwide by country 2021*. Statista. <https://www.statista.com/statistics/1222815/loneliness-among-adults-by-country/>
101. Wada, K., Shibata, T., Musha, T., & Kimura, S. (2008). Robot therapy for elders affected by dementia. *IEEE Engineering in Medicine and Biology Magazine*, 27(4), 53–60. <https://doi.org/10.1109/memb.2008.919496>
102. Wallach, W., & Allen, C. (2010). *Moral machines: Teaching robots right from wrong*. Oxford University Press.
103. Weizenbaum, J. (1966). ELIZA—A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1), 36–45. <https://doi.org/10.1145/365153.365168>
104. Nirmalie Wiratunga, Cooper, K., Anjana Wijekoon, Chamath Palihawadana, Mendham, V., Reiter, E., & Martin, K. E. (2020). *FitChat: Conversational artificial intelligence interventions for encouraging physical activity in older adults*.
105. Wu, Y.-H., Fassert, C., & Rigaud, A.-S. (2012). Designing robots for the elderly: Appearance issue and beyond. *Archives of Gerontology and Geriatrics*, 54(1), 121–126. <https://doi.org/10.1016/j.archger.2011.02.003>
106. Xie, T., & Pentina, I. (2022). Attachment theory as a framework to understand relationships with social Chatbots: A case study of Replika. [scholarspace.manoa.hawaii.edu. http://hdl.handle.net/10125/79590](http://hdl.handle.net/10125/79590)
107. Xie, T., Iryna Pentina, & Hancock, T. (2023). *Friend, mentor, lover: Does chatbot engagement lead to psychological dependence?* <https://doi.org/10.1108/josm-02-2022-0072>
108. Zang, Y., Zhang, F., Di, C., & Zhu, D. (2015). Advances of flexible pressure sensors toward artificial intelligence and health care applications. *Materials Horizons*, 2(2), 140–156. <https://doi.org/10.1039/c4mh00147h>

The SEARCH for AI-Informed Wellbeing Education: A Conceptual Framework



Kelly-Ann Allen, Margaret L. Kern, Joseph Crawford, Michael Cowling, Duyen Vo, and Lea Waters

Abstract The rapid advancement of generative artificial intelligence (AI) and large language model (LLM) technologies, such as ChatGPT-4 and Bard, has the potential to significantly change wellbeing education. This Chapter explores the applications of generative AI technologies in wellbeing education, with a focus on how chatbots and similar can be used to cultivate wellbeing through the SEARCH framework. For clarity, the SEARCH framework focuses on developing Strengths, Emotional management, Attention and awareness, Relationships, Coping, and Habits and goals. We begin by presenting the potential benefits of incorporating generative AI in wellbeing education. Next, by employing the SEARCH framework as a model of wellbeing education, the Chapter broadly conceptualises how AI technologies can be used to teach and explore the SEARCH components. The potential impact of AI-enhanced wellbeing education on teaching and learning practice—with implications for preparing teachers with ethical considerations and practical knowledge for using such technology—are also discussed.

Keywords Wellbeing education · Artificial intelligence · ChatGPT · Large language models · Chatbot · Student wellbeing

K.-A. Allen · D. Vo
Monash University, Melbourne, Australia

K.-A. Allen · M. L. Kern · L. Waters
University of Melbourne, Melbourne, Australia

J. Crawford (✉)
University of Tasmania, Hobart, Australia
e-mail: joseph.crawford@utas.edu.au

M. Cowling
Central Queensland University, Melbourne, Australia

1 Introduction

The mental health and wellbeing of young people continue to be crucial topics of discussion worldwide, as the prevalence of mental health disorders increases. According to the World Health Organisation (WHO), one in seven 10 to 19-year olds experiences a mental disorder, with depression and anxiety among the most common causes [56]. Youth suicide has also been a grim concern globally. The Australian Institute of Health and Welfare reported that suicide was the leading cause of death among Australians aged between 15 and 24 [1]. In the United States of America, nearly one in three teen girls seriously considered attempting suicide [6] while across Europe, adolescent lives are lost every day due to mental health issues [44].

Wellbeing education, which broadly refers to programs and interventions that promote emotional, social and mental wellbeing in educational contexts [3, 55], is an essential component of preventing and addressing mental health disorders. Schools provide an ideal setting to teach and develop the necessary skills and mindsets in young people to manage their emotions, develop healthy relationships, and build resilience [8, 30, 52]. Studies have shown that school-based wellbeing programs are significantly linked to student wellbeing [8, 17, 47]. For example, a systematic review of children and adolescent mental health interventions in schools and communities, with a focus on social interactions, showed that interventions had positive outcomes on both treating and preventing depression and anxiety [17]. Furthermore, Waters' [47] review of school-based positive psychology interventions, which build on positive factors such as gratitude, hope and resilience, showed an increase in life satisfaction, optimism, relationships and indeed, academic performance among adolescents.

Traditionally, student wellbeing education programs have been developed as brief, face-to-face interventions implemented by school teachers [2, 15, 31, 47]. Technology and the internet have expanded learning and teaching opportunities, providing students and teachers more flexibility, equity and potential for improving user engagement [14, 18]. For example, the *Bio-Dash Program* was developed to incorporate the use of dynamic technology in gamification tasks to help students learn how to manage stress and build resilience to maximise performance [45]. Consisting of 34 brief online modules and a unique biofeedback tool, which monitors physiological functions, evaluations from students indicated that the technological aspects of the program made learning about wellbeing strategies more engaging and enjoyable [54]. Another successful use of technology in wellbeing education is *Smiling Mind*, an application (app) based program offering a variety of modules in the training and application of mindfulness [24]. Although available to the wider population, the program offers diverse options for schools to develop students' social and emotional learning in a whole-school context. Research conducted with more than 1800 students and 100 teachers in Victoria, Australia suggested participation in the program improved mental health outcomes (e.g., sleep quality and classroom concentration) and reduced classroom disruptions, especially in at-risk students [23].

Technology and the internet have helped boost young people's wellbeing by offering engaging lessons and practise tools [27, 45, 54], and equipping teachers with accessible and creative tools to be able to integrate wellbeing strategies into everyday lessons and whole-school curriculum. The rapid advancement of artificial intelligence (AI) presents new and exciting possibilities to further discover, create and develop the wellbeing education 'toolkit'.

The application of AI technologies in education is already proving to be transformative, as educators discover the potential to impact teaching and learning practices [7, 10], with chatbot opportunities for fostering student wellbeing [33]. One of the more popular applications of AI [19], ChatGPT, was first released in November 2022, with its most recent stable 'ChatGPT-4' released in March 2023. As a type of Generative AI technology, ChatGPT deploys large language modelling (LLM) to make conversational responses to reasonably complex prompts. The tool can also create case studies, simulations, and provide tailored feedback (<https://chat.openai.com/chat>).

While its history is short, ChatGPT's influence on contemporary education has been significant [10, 41]. In the context of wellbeing education, AI-enhanced technologies such as ChatGPT offer new opportunities for promoting wellbeing. As wellbeing education continues to gain momentum in educational settings [35], it is crucial to understand the implications of AI advancements on teaching and learning practices, as well as implications for student and teacher wellbeing.

But how can generative artificial intelligence be used to create various learning experiences that support *students flourishing*, and do so in way that embeds academic integrity [38]? We argue that generative artificial intelligence can be used to: tailor strengths feedback, generate wellbeing strategies, create age-appropriate simulations where students can practice wellbeing approaches before having to use them in a real situation, run mindfulness and relaxation activities, analyse group dynamics using case studies, develop goal-setting guidelines, and provide guided resources based on the area of wellbeing that a student wishes to work upon. This chapter begins by presenting the benefits of incorporating generative artificial intelligence in wellbeing education. Then, through exploring each component of the wellbeing framework SEARCH, we provide suggestions on how to incorporate AI-driven capabilities. To summarise, we present a useful visual summary, which brings together the ideas.

2 Why Use Generative AI in Wellbeing Education?

Generative AI has the potential to revolutionise wellbeing education for both students and teachers. First, it has the ability to efficiently and effectively create personalised student-centred content that is tailored to individual students' strengths, needs and interests [37]. By analysing data specific to the individual student (e.g., learning style, character profiles, mental health plan and goals), generative AI algorithms can identify areas where each student may need additional support or resources. This can lead to a more targeted and effective approach to wellbeing education, as students

are given more relevant and meaningful learning tools and information. This, in turn, is more likely to motivate and engage students with the content [29].

Second, generative AI can help to create engaging and creative learning experiences that capture students' attention and promote active participation. Together with a whole-school approach to student learning, buy-in has been identified as a central factor in the long-term success of any wellbeing program [46]. By incorporating interactive elements, generative artificial intelligence can make learning about wellbeing more fun and memorable, which encourages 'buy-in' [14].

Third, generative AI can be used to monitor and track student wellbeing over time. Teachers, school leaders, wellbeing teams, school counsellors and psychologists can use generative AI to analyse data collected from student wellbeing surveys on factors such as sleep patterns, exercise habits and stress levels. This data can then be used to identify students who may be struggling and provide targeted support and intervention.

Finally, generative AI can also be used to support teacher training and professional development. Using AI-enhanced feedback and coaching can provide educators with targeted resources and activities based on their unique strengths, interests, and learning needs. For example, an AI-powered learning platform could recommend articles, videos, and other resources.

Educators can also elicit teaching strategies, lesson plans, and interventions, incorporating wellbeing approaches effective in promoting student wellbeing and engagement. AI technology also has the capacity to authentically guide educators with skills to embed wellbeing education principles into existing curriculum. For example, we asked ChatGPT how an educator might integrate relationship building and empathy into a typical literacy lesson:

To integrate empathy into a literacy lesson, educators can encourage students to empathize with the characters in the story and connect their experiences to their own lives.

For example, when reading a novel or story, educators can guide students to think about the emotions and experiences of the characters and how they might relate to their own lives. Educators can ask open-ended questions such as, "How do you think the character felt when that happened?" or "Have you ever experienced something similar to what the character is going through?"

Given it has been argued that some of the barriers to implementing and sustaining school-based health and wellbeing programs and interventions include staff engagement, quality of materials and training, and time [32, 42], using generative artificial intelligence can help build teachers' own skills and competencies in wellbeing education [26, 38].

3 The SEARCH for AI-Driven Wellbeing Education

In the last section, we outlined reasons why using generative AI can enhance wellbeing education. This section aims to specifically show how teachers can use ChatGPT to help students practise and develop wellbeing skills. Through employing

the evidenced-based SEARCH framework [49, 51, 53], we explore how a range of AI-generated applications of ChatGPT-4 (e.g., simulations, tailored feedback, mindfulness scripts etc.) [33] can be used to bolster the six SEARCH pathways for student wellbeing: strengths, emotional management, attention and awareness, relationships, coping, habits and goals.

The SEARCH pathway is an evidence-based model used by teachers to more intentionally seek to build student wellbeing. Based on a large-scale review of positive psychology studies and then testing the validity of wellbeing themes identified, the framework has been found to boost stress-related growth for students in Australia during lockdown [50] and prevent students in Hong Kong from suffering declines in wellbeing during the pandemic [51]. The SEARCH framework has been the basis of fostering student wellbeing through traditional lessons and classroom interventions [52] but has not yet been used with an AI approach.

3.1 Strengths

Strengths can be broadly defined as skills and behaviors that feel natural and enjoyable to an individual, and allow them to perform at their best [21, 48]. By maximising individuals' potential, it has been suggested that cultivating character strengths contributes to both "psychological and physiological wellbeing and functioning" [28]. Indeed, character-strengths interventions with adolescents at schools have been shown to increase life satisfaction [39] and predict subjective wellbeing [20].

Typically, interventions at schools involve four areas of learning [28]. The first provides students with a theoretical understanding. For example, students explore definitions and descriptions about different strengths. Second, students are encouraged to identify or spot their individual strengths, either utilising a questionnaire or answering open-ended questions [28, 40]. Next, students are encouraged to put their strengths into action in various ways, and finally, interventions often involve a self-reflection exercise (Fig. 1).

Furthermore, the platform can also provide strategies for cultivating strengths. For example, students or teachers can provide ChatGPT with a profile of their identified strengths and ask it to create an action plan with daily activities. Concerning self-reflection tools, ChatGPT can tailor and generate questions as prompts drawing on individual student's strengths and profiles (e.g., age, year level).

3.2 Emotional Management

Emotional regulation skills are crucial to support mental health and wellbeing [22]. AI technologies hold the potential to enhance emotional regulation competencies in children and students in a variety of ways, (e.g., simulations, activities, strategies),

DR I want to do a "strength spotting" exercise with you. You ask me a question and I will answer your question. From my response, I want you to identify my strengths using the VIA Classification of Character Strengths. You ask the question. I respond. Do you understand?



Yes, I understand. Let's get started with the exercise.



Question: Can you think of a situation where you felt particularly proud of yourself or accomplished, either at work or in your personal life?

Please respond when you're ready.



I feel proud of myself when I have finished a project. The project might be small like writing a short opinion piece on wellbeing education. I enjoy writing but also helping others to see the benefits of using new tools and technologies.



Thank you for sharing. Based on your response, it seems that you may possess strengths such as:

1. Love of Learning: Your interest in writing and well-being education suggests a love of learning and a desire to acquire new knowledge and skills.
2. Creativity: Your enjoyment of writing and helping others see the benefits of new tools and technologies suggests strength in creativity, as you may have a natural talent for generating new ideas or approaches.
3. Persistence: Your ability to finish a project, no matter how small, suggests strength in persistence, as you may have the ability to stick with a task until it is completed.
4. Self-regulation: Your ability to focus on and complete a project suggests strength in self-regulation, as you may have the ability to manage your emotions, thoughts, and behaviors in order to achieve your goals.
5. Leadership: Your interest in helping others see the benefits of new tools and technologies suggests strength in leadership, as you may have a natural talent for guiding and inspiring others.
6. Perspective: Your interest in well-being education suggests strength in perspective, as you may have the ability to think deeply about complex issues and consider multiple viewpoints in order to make informed decisions.

Based on your response, these are some strengths that you may possess according to the VIA Classification of Character Strengths. I hope this helps, and let me know if you have any further questions.

Regenerate response

Send a message...



Fig. 1 Using ChatGPT to engage in a strength-spotting exercise

while simultaneously considering the developmental needs of the target population. For example, emotion management in early childhood may incorporate understanding and identifying feelings, while emotion management in secondary school might involve skills such as rational thinking and cognitive reframing. AI-generative technologies have utility in considering factors like age appropriateness, while also providing educators with the tools to build emotion management competencies in the children or young people they work with [43].

Specifically, through AI, educators can create interactive simulation activities that allow students to safely practice or consider emotional regulation responses. For example, AI could simulate scenarios tailored for adolescents, such as needing to manage emotions following conflict with a friend, experiencing anxiety before a test, or managing frustration in a group project generating various options for how a young person can respond and providing feedback on their choices related to anticipated outcomes. For instance, we asked ChatGPT, to create a scenario for a 15 year old managing anxiety before a test.

Scenario: You have a big exam coming up in your favorite subject, but you're feeling very anxious and stressed about it. You're having trouble focusing on your studying, and your mind keeps wandering to worst-case scenarios about the test.

Option 1: Positive Self-Talk and Relaxation Techniques

Take a few deep breaths and remind yourself that you've studied hard and are well-prepared for the exam. Use positive self-talk to remind yourself of your strengths and capabilities. Try relaxation techniques such as meditation or progressive muscle relaxation to help calm your mind and body.

Option 2: Seek Social Support

Reach out to a friend or family member for emotional support. Talk to them about how you're feeling and ask for their encouragement and reassurance. If possible, find a study group or tutor who can help you prepare for the exam and offer feedback and support.

Option 3: Reframe Negative Thoughts

Identify any negative thoughts or beliefs that are contributing to your anxiety and challenge them with more positive and realistic thoughts. For example, if you're thinking "I'm going to fail this exam," reframe it as "I've studied hard and am prepared for the exam, and I will do my best."

3.3 Attention and Awareness

Attention and awareness enable students to develop self-awareness and focus [5]. ChatGPT-4 can be used to guide students through mindfulness exercises, such as breath awareness, body scans, and loving-kindness meditations [25]. These practices can help students focus attention on the present moment and cultivate a non-judgmental awareness of their thoughts, emotions, and bodily sensations [4].

We asked ChatGPT to provide an example of a body scan exercise that an educator could read out to a class of four-year-olds:

Imagine you're a small seed that's just been planted in the ground. You're safe and warm, and you can feel the soil all around you. Take a deep breath in, and as you exhale, feel your roots start to grow down into the soil. Wiggle your toes and imagine your roots stretching out further and further.

As your roots grow, you start to feel taller and stronger. Your stem shoots up towards the sky, and your leaves unfurl to soak up the sunshine. Take another deep breath in, and as you exhale, imagine the sun's rays warming your leaves and filling you with energy.

Now, let's scan through your body and see how it feels. Wiggle your fingers and toes, and feel the energy flowing through your body. Take a deep breath in, and as you exhale, let your body feel heavy and relaxed.

Focus on your head and imagine a gentle breeze blowing through your hair. Move down to your eyes and feel them relax and soften. Move down to your nose and take a deep breath in, smelling the fresh air all around you.

Move down to your mouth and imagine a big smile spreading across your face. Feel the warmth and joy that comes from smiling. Move down to your shoulders and imagine them relaxing and dropping down away from your ears.

Move down to your arms and feel them heavy and relaxed. Wiggle your fingers and feel the energy flowing through them. Move down to your legs and feel them strong and steady, rooted in the ground like a tree.

Finally, take a deep breath in and imagine your whole body filled with warmth and light. As you exhale, imagine all your worries and stress flowing out of your body and into the ground. You are now a strong, healthy, and relaxed tree, ready to face the day.

By framing the body scan exercise in a novel and imaginative way, educators can make this activity engaging and fun for young children. This exercise also promotes mindfulness and relaxation, which can help young children develop self-awareness and manage stress and anxiety.

3.4 Relationships

AI technologies can play a critical role in assisting teachers to effectively teach relationship skills through opportunities to practice conflict resolution skills and by fostering an inclusive learning environment for which students can build a sense of belonging to [9]. One potential innovative application of ChatGPT-4 is to promote relationship skills through the analysis of classroom interactions and the provision of targeted feedback to educators. By assessing teacher-to-student and peer-to-peer interactions, AI-driven tools can identify areas for improvement, provide recommendations for enhancing communication and understanding within the learning environment, and analyse group dynamics.

We asked ChatGPT how it can be used to cultivate relationships and relationship skills of children and young people?:

Real-time feedback: ChatGPT-4 can provide real-time feedback to teachers on their interactions with students. For example, the tool can analyze the tone and language used by the teacher during class discussions and provide suggestions on how to modify their approach to better engage students and foster positive relationships.

Student feedback: ChatGPT-4 can also provide feedback to students on their communication skills and interactions with peers. For example, the tool can analyze student responses during group work and provide suggestions on how to improve collaboration and communication with their classmates.

Group dynamics analysis: ChatGPT-4 can analyze group dynamics and provide recommendations to teachers on how to foster positive interactions and relationships within the classroom. For example, the tool can identify students who are isolated or excluded from group activities and provide suggestions on how to integrate them into the group.

3.5 Coping

For teachers, AI technologies offer major advantages in developing rapid information that draws from the best available evidence to enhance their ability to teach coping strategies to respond to different situations and stressors dynamically as needs arise [16]. For example, a teacher may consider teaching coping skills to a class that has been reporting several difficulties during break-times of children leaving others out of play. If a teacher asks GPT4-chat to generate a case study about social isolation in elementary/primary school, the AI tool will generate a case study and lesson plan similar to the one below:

Case Study:

Sarah is an 8-year-old girl who has been feeling sad and lonely lately. She has a group of friends at school, but sometimes they leave her out or choose to play with other kids. Sarah feels like she doesn't fit in and is not sure what to do about it. She often comes home feeling upset and doesn't know how to cope with her feelings.

Instructions:

Begin by introducing the concept of coping skills and asking the children to brainstorm different coping strategies they know. Write these on the board.

Explain that there are productive and non-productive coping strategies. Productive strategies help us feel better and solve problems, while non-productive strategies may make us feel better temporarily but do not solve the underlying problem.

Ask the children to identify which coping strategies on the board are productive and which are non-productive. Discuss why some strategies may be more helpful than others.

Next, present Sarah's case study to the group and ask them to identify how she might be feeling and what non-productive coping strategies she might be using.

Divide the class into small groups and ask them to come up with three productive coping strategies that Sarah could use when she feels left out by her friends.

Each group will then share their ideas with the class and discuss why they believe their strategies are productive.

As a group, come up with a list of productive coping strategies for Sarah to use when she is feeling sad or lonely.

Finally, ask each child to choose one productive coping strategy that they will try to use the next time they face a difficult situation.

Questions:

- What are some productive coping strategies you can use when you feel left out or lonely?
- Why do you think some coping strategies are more helpful than others?
- How do you think Sarah might be feeling, and what non-productive coping strategies might she be using?
- How can you use productive coping strategies to feel better and solve problems?
- What coping strategy will you try to use the next time you face a difficult situation?

3.6 Habits and Goals

AI technologies can be used to enhance teacher understanding of cultivating healthy habits and setting realistic goals. By presenting teachers with guidance on evidence-based methodologies for teaching healthy habits and goal-setting techniques, AI tools can support teachers to incorporate goal-setting processes more readily into their teaching. AI-driven tools can then provide customized recommendations, feedback, and reinforcement to help teachers and students create goals and maintain motivation towards their goals.

A major strength of AI-driven tools is their unique ability to meet the needs of diverse student populations, through responses that can target individual differences. For example, a teacher could use AI to provide feedback on the goal, “I want to exercise more” to a 12 year old who uses a wheelchair, enjoys basketball, and wants to improve their engagement in regular exercise. Using a SMART (Specific, Measurable, Achievable, Relevant, and Time-bound) methodology and taking into consideration the unique needs of the student, the goal can be improved using AI:

Specific: “I want to exercise more by doing wheelchair basketball for at least 20 minutes a day, four times a week.”

Measurable: “I will keep a log of my wheelchair basketball sessions, including the duration and how I felt after each session.”

Achievable: “I will start with a lower intensity and gradually increase the time and intensity of my wheelchair basketball sessions as my fitness improves.”

Relevant: “I want to improve my strength, flexibility, and overall health so that I can continue to do the things I love, like spending time with my friends and family.”

Time-bound: “I will work towards my goal of exercising through wheelchair basketball for the next three months and aim to increase the frequency and duration of my sessions over time.”

3.7 Summary of AI-Enhanced SEARCH Framework

Through a targeted application of the six SEARCH pathways for student wellbeing: strengths, emotional management, attention and awareness, relationships, coping,

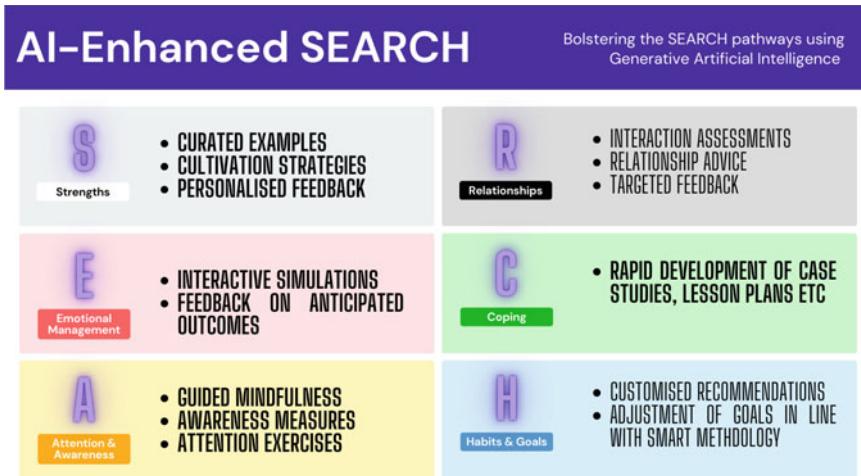


Fig. 2 AI-enhanced SEARCH framework

habits and goals, this chapter has demonstrated examples of how ChatGPT (or Generative artificial intelligence) can be used to support student wellbeing. Our overall framework is summarised below (Fig. 2).

4 Conclusion

The rapid advancement of AI technologies offers transformative opportunities for wellbeing education. By employing the SEARCH framework as an evidence-based model that builds wellbeing [49, 53], this chapter has explored the potential applications of ChatGPT-4 in shaping AI-driven wellbeing education to create personalized learning experiences to enhance various components of student flourishing.

Future research in this area could focus on evaluating the long-term impact of AI-driven wellbeing education on student wellbeing, as well as exploring the potential of emerging AI technologies in addressing specific challenges and opportunities within the field, in particular considering how use changes what we consider authoring and ownership of resources by teachers and students [11]. By continuing to innovate and adapt, AI-driven wellbeing education holds the promise of enhancing teaching and learning practices, ultimately contributing to the flourishing of students and educators alike.

Conflict of Interest The author(s) disclose that they have no actual or perceived conflicts of interest. The authors disclose that they have not received any funding for this manuscript beyond resourcing for academic time at their respective university. ChatGPT was used for idea and example generation in this Chapter only.

References

1. Australian Institute of Health and Welfare. (2023). Deaths in Australia. <https://www.aihw.gov.au/reports/life-expectancy-deaths/deaths-in-australia/contents/leading-causes-of-death>
2. Bernard, M. E., & Walton, K. (2011). The effect of You Can Do It! Education in six schools on student perceptions of wellbeing, teaching, learning and relationships. *Journal of Student Wellbeing*, 5(1), 22–37.
3. Beyond Blue. (2023). Be you: Growing a mentally healthy generation. <https://beyou.edu.au/>
4. Bishop, S. R., Lau, M., Shapiro, S., Carlson, L., Anderson, N. D., Carmody, J., Segal, Z. V., Abbey, S., Speca, M., Velting, D., & Devins, G. (2004). Mindfulness: A proposed operational definition. *Clinical Psychology: Science and Practice*, 11(3), 230–241. <https://doi.org/10.1093/clipsy.bph077>
5. Brown, K. W., Ryan, R. M., & Creswell, J. D. (2007). Mindfulness: Theoretical foundations and evidence for its salutary effects. *Psychological Inquiry*, 18(4), 211–237. <https://doi.org/10.1080/10478400701598298>
6. Centers for Disease Control and Prevention. (2023). *U.S. teen girls experiencing increased sadness and violence*. <https://www.cdc.gov/media/releases/2023/p0213-yrbs.html>
7. Chen, L., Chen, P., & Lin, Z. (2020). Artificial intelligence in education: A review. *IEEE Access*, 8, 75264–75278.
8. Cilar, L., Štiglic, G., Kmetec, S., Barr, O., & Pajnkihar, M. (2020). Effectiveness of school-based mental well-being interventions among adolescents: A systematic review. *Journal of Advanced Nursing*. Blackwell Publishing Ltd. <https://doi.org/10.1111/jan.14408>
9. Crawford, J., Allen, K. A., Sanders, T., Baumeister, R., Parker, P., Saunders, C., & Tice, D. (2023). Sense of belonging in higher education students: An Australian longitudinal study from 2013 to 2019. *Studies in Higher Education*, pp. 1–15.
10. Crawford, J., Cowling, M., & Allen, K. A. (2023). Leadership is needed for ethical ChatGPT: Character, assessment, and learning using artificial intelligence (AI). *Journal of University Teaching & Learning Practice*, 20(3). <https://doi.org/10.53761/1.20.3.02>
11. Crawford, J., Cowling, M., Ashton-Hay, S., Kelder, J-A., Middleton, R. & Wilson, G. (2023). Artificial Intelligence and Authorship Editor Policy: ChatGPT, Bard Bing AI, and beyond. *Journal of University Teaching and Learning Practice*, 20(5). <https://doi.org/10.53761/1.20.5.01>
12. Creswell, J. D. (2017). Mindfulness interventions. *Annual Review of Psychology*, 68, 491–516. <https://doi.org/10.1146/annurev-psych-042716-051139>
13. Dweck, C. S. (2006). *Mindset: The new psychology of success*. Random House.
14. Francis, J., Vella-Brodrick, D., & Chyuan-Chin, T. (2021). Effectiveness of online, school-based Positive Psychology Interventions to improve mental health and wellbeing: A systematic review. *International Journal of Wellbeing*, 11(4), 44–67. <https://doi.org/10.5502/ijw.v11i4.1465>
15. Froh, J. J., Kashdan, T. B., Ozimkowski, K. M., & Miller, N. (2009). Who benefits the most from a gratitude intervention in children and adolescents? Examining positive affect as a moderator. *Journal of Positive Psychology*, 4(5), 408–422. <https://doi.org/10.1080/17439760902992464>
16. Frydenberg, E., Deans, J., & O'Brien, K. A. (2012). *Developing everyday coping skills in the early years: Proactive strategies for supporting social and emotional development*. Bloomsbury.
17. Garcia-Carrion, R., Villarejo, B. C., & Villardón-Gallego, L. (2019). Children and adolescents mental health: A systematic review of interaction-based interventions in schools and communities. In *Frontiers in Psychology* (Vol.10, Issue APR). Frontiers Media S.A. <https://doi.org/10.3389/fpsyg.2019.00918>
18. Garrido, S., Millington, C., Cheers, D., Boydell, K., Schubert, E., Meade, T., & Nguyen, Q. V. (2019). What works and what doesn't work? A systematic review of digital mental health interventions for depression and anxiety in young people. In *Frontiers in Psychiatry* (Vol. 10). Frontiers Media S.A. <https://doi.org/10.3389/fpsy.2019.00759>

19. George, A. S., & George, A. H. (2023). A review of ChatGPT AI's impact on several business sectors. *Partners Universal International Innovation Journal*, 1(1), 9–23. <https://doi.org/10.5281/zendodo.7644359>
20. Gillham, J., Adams-Deutsch, Z., Werner, J., Reivich, K., Coulter-Heindl, V., Linkins, M., et al. (2011). Character strengths predict subjective well-being during adolescence. *The Journal of Positive Psychology*, 6, 31–44. <https://doi.org/10.1080/17439760.2010.536773>
21. Govindji, R., & Linley, P. A. (2007). Strengths use, self-concordance and well-being: Implications for strengths coaching and coaching psychologists. *International Coaching Psychology Review*, 2(2), 143–153. <https://doi.org/10.1037/t01038-000>
22. Gross, J. J. (2015). Emotion regulation: Current status and future prospects. *Psychological Inquiry*, 26(1), 1–26. <https://doi.org/10.1080/1047840X.2014.940781>
23. Hart, P. (2016). Establishing an evidence base for the Smiling Mind Education Program: Preliminary results from a large study conducted by Deakin University and Insight SRC.
24. Johnson, D., Hides, L., Kavanagh, D. J., Zelenko, O., Stoyanov, S. R., Cockshaw, W., Staneva, A., & Wilson, H. (2016). Smiling Mind-Game on: A gamified mindfulness meditation program for young people.
25. Kabat-Zinn, J. (2003). Mindfulness-based interventions in context: Past, present, and future. *Clinical Psychology: Science and Practice*, 10(2), 144–156. <https://doi.org/10.1093/clipsy.bpg016>
26. Kelly, A., Sullivan, M., & Strampel, K. (2023). Generative artificial intelligence: University student awareness, experience, and confidence in use across disciplines. *Journal of University Teaching & Learning Practice*, 20(6), 12. <https://doi.org/10.53761/1.20.6.12>
27. Lahtinen, O., & Salmivalli, C. (2020). An effectiveness study of a digital mindfulness-based program for upper secondary education students. *Mindfulness*, 11(11), 2494–2505. <https://doi.org/10.1007/s12671-020-01462-y>
28. Lavy, S. (2020). A review of character strengths interventions in twenty-first-century schools: Their importance and how they can be fostered. *Applied Research in Quality of Life*, 15(2), 573–596. <https://doi.org/10.1007/s11482-018-9700-6>
29. Li, K. C., Wong, B. T. M. (2022). Research landscape of smart education: A bibliometric analysis. *Interactive Technology and Smart Education*, 19(1), 3–19.
30. Marques, S. C., Lopez, S. J., & Pais-Ribeiro, J. L. (2011). Building hope for the future: A program to foster strengths in middle-school students. *Journal of Happiness Studies*, 12(1), 139–152. <https://doi.org/10.1007/s10902-009-9180-3>
31. Marques, S. C., Pais-Ribeiro, J. L., & Lopez, S. J. (2011). The role of positive psychology constructs in predicting mental health and academic achievement in children and adolescents: A two-year longitudinal study. *Journal of Happiness Studies*, 12(6), 1049–1062.
32. Moore, A., Stapley, E., Hayes, D., Town, R., Deighton, J. (2022). Barriers and facilitators to sustaining school-based mental health and wellbeing interventions: A systematic review. *International Journal of Environmental Research and Public Health*, 19(6), 3587. <https://doi.org/10.3390/ijerph19063587>; PMID: 35329276; PMCID: PMC8949982.
33. O'Dea, X. C., & O'Dea, M. (2023). Is artificial intelligence really the next big thing in learning and teaching in higher education? A conceptual paper. *Journal of University Teaching and Learning Practice*, 20(5). <https://doi.org/10.53761/1.20.5.06>
34. Ouyang, F., Zheng, L., & Jiao, P. (2022). Artificial intelligence in online higher education: A systematic review of empirical research from 2011 to 2020. *Education and Information Technologies*, 27(6), 7893–7925. <https://doi.org/10.1007/s10639-022-10925-9>
35. Owens, R. L., & Waters, L. (2020). What does positive psychology tell us about early intervention and prevention with children and adolescents? A review of positive psychological interventions with young people. *The Journal of Positive Psychology*, 15(5), 588–597. <https://doi.org/10.1080/17439760.2020.1789706>
36. Park, N., Peterson, C., & Seligman, M. E. (2004). Strengths of character and well-being. *Journal of Social and Clinical Psychology*, 23(5), 603–619. <https://doi.org/10.1521/jscp.23.5.603.50748>

37. Patrick, S., Kennedy, K., & Powell, A. (2013). *Mean what you say: Defining and integrating personalized, blended, and competency education*. Aurora Institute. <https://aurora-institute.org/wp-content/uploads/mean-what-you-say-1.pdf>
38. Perkins, M. (2023). Academic Integrity considerations of AI large language models in the post-pandemic era: ChatGPT and beyond. *Journal of University Teaching and Learning Practice*, 20(2), 07. <https://doi.org/10.53761/1.20.20.07>
39. Proctor, C., Tsukayama, E., Wood, A. M., Maltby, J., Eades, J. F., & Linley, P. A. (2011). Strengths Gym: The impact of a character strengths-based intervention on the life satisfaction and well-being of adolescents. *The Journal of Positive Psychology*, 6(5), 377–388. <https://doi.org/10.1080/17439760.2011.594079>
40. Quinlan, D., Vella-Brodrick, D. A., Gray, A., & Swain, N. (2019). Teachers matter: Student outcomes following a strengths intervention are mediated by teacher strengths spotting. *Journal of Happiness Studies*, 20(8), 2507–2523. <https://doi.org/10.1007/s10902-018-0051-7>
41. Rudolph, J., Tan, S., & Tan, S. (2023). ChatGPT: Bullshit spewer or the end of traditional assessments in higher education? *Journal of Applied Learning and Teaching*, 6(1), 1–22. <https://doi.org/10.37074/jalt.2023.6.1.9>
42. Shankland, R., & Rosset, E. (2017). Review of brief school-based positive psychological interventions: A taster for teachers and educators. *Educational Psychology Review*, 29(2), 363–392. <https://doi.org/10.1007/s10648-016-9357-3>
43. Su, J., & Yang, W. (2022). Artificial intelligence in early childhood education: A scoping review. *Computers and Education: Artificial Intelligence*, Article 100049. <https://doi.org/10.1016/j.caai.2022.100049>
44. UNICEF. (2021). *State of the world's children 2021: On my mind promoting, protecting and caring for children's mental health*.
45. Vella-Brodrick, D. (2019). How technology is boosting our young people's wellbeing. Pursuit.
46. Vella-Brodrick, D. A., Chin, T-C., & Rickard, N. S. (2019). Examining the processes and effects of an exemplar school-based well-being approach on student competency, autonomy and relatedness. *Health Promotion International*, daz115.
47. Waters, L. (2011). A review of school-based positive psychology interventions. *Australian Educational and Developmental Psychologist*, 28(2), 75–90. <https://doi.org/10.1375/aedp.28.2.75>
48. Waters, L. (2017). *The strength switch: How the new science of strength-based parenting can help your child and your teen to flourish*. Penguin Random House.
49. Waters, L. (2019). Searching for wellbeing in schools: A new framework to guide the science of positive education. *Journal of Educational Psychological Research*, 1(2), 1–8. <https://doi.org/10.33140/JEPR.01.02.02>
50. Waters, L., Allen, K., & Arslan, G. (2021). Adversarial growth in adolescents returning to school after Covid-19 school closure. *Frontiers*. <https://doi.org/10.3389/fpsyg.2021.643443>
51. Waters, L., Bullock, J., & Loton (2023). When adversity strikes: Maintaining student and teacher wellbeing during the Covid-19 pandemic. *Journal of Psychology and Behavioral Science*.
52. Waters, L., & Johnstone, A. (2022). Embedding well-being into school: A case study of positive education before and during COVID-19 lockdowns. *Journal of School and Educational Psychology*, 2(2), 60–77. <https://doi.org/10.47602/josep.v2i2.15>
53. Waters, L., & Loton, D. (2019). SEARCH: A meta-framework and review of the field of positive education. *International Journal of Applied Positive Psychology*, 4(1–3), 1–46. <https://doi.org/10.1007/s41042-019-00017-4>
54. West, M., Patrick, K., & Vella-Brodrick, D. (2022). SMART technology has an important role to play in making learning about wellbeing in schools engaging and real for students. In *Handbook of Positive Psychology in Schools: Supporting Process and Practice* (pp. 511–524). Taylor and Francis. <https://doi.org/10.4324/9781003013778-41>

55. White, M. A. (2021). A decade of positive education and implications for initial teacher education: A narrative review. *Australian Journal of Teacher Education*, 46(3), 74–90. <https://doi.org/10.3316/aeipt.228961>
56. World Health Organisation. (2021). Mental health of adolescents. <https://www.who.int/data/gho/data/major-themes/health-and-well-being>

Generative AI to Understand Complex Ecological Interactions



Hirn Johannes, Sanz Verónica, and Verdú Miguel

Abstract The recent use of Generative AI (GenAI) techniques in Ecology has provided insights into predicting species co-occurrence patterns, specifically in water-limited ecosystems where multispecific plant clumps grow sparsely. In particular, these patterns have been employed to elucidate the mechanisms governing the assembly of plant communities in the context of Southeastern Spain. We discuss how the important concepts of transfer learning, and data augmentation take on slightly different meanings in this context, as compared to their usual application in Computer Vision. In particular, using transfer learning, the same models have been successfully applied to other plant communities in another semi-arid region of Spain and of tropical Mexico, opening the door to a specific kind of data augmentation by combining data sets from disparate communities. Beyond that, we also discuss the use of GenAI for synthetic data, and for predictions that can be of practical use when replanting vegetation in degraded environments, with an eye to biodiversity.

Keywords Variational encoders · Transfer learning · Data augmentation · Synthetic data · Restoration ecology

1 Introduction

As is the case for many areas of life and science, Ecology can make use of techniques from AI. In fact, given that ecological communities are complex systems by nature (with large numbers of interacting variables) they lend themselves well to Machine

H. Johannes (✉) · V. Miguel

Centro de Investigaciones Sobre Desertificación (CIDE, CSIC-Universidad de Valencia-Generalitat Valenciana), 46113 Valencia, Spain
e-mail: johannes.hirn@ext.uv.es

S. Verónica

Instituto de Física Corpuscular (IFIC, Universidad de Valencia-CSIC), 46980 Valencia, Spain

Department of Physics and Astronomy, University of Sussex, Brighton BN1 9QH, UK

Learning (ML) approaches that can help in extracting the relevant patterns from the proverbial haystack of possible variables [5, 10, 21].

In this Chapter, we introduce the reader to some issues in Ecology that can be tackled using Generative Artificial Intelligence (GenAI). We first expose a few fundamental notions about Ecology as it pertains to plant species and their interactions, then move on to describe examples of the uses of GenAI in that field.

In the era of Big Data, Ecology is faced with one difficulty: the timescales and costs associated with data collection. While some ecological data can be gathered remotely or automatically—animal tracking via ground-based antennas or satellite, large-scale satellite imaging of vegetation health, identification and measurements of species and individuals using camera traps or Unmanned Autonomous Vehicles (UAVs)—there are large and important subfields of Ecology for which these automated methods do not apply, and one such data-starved field is Restoration Ecology.

Restoration Ecology aims to rehabilitate ecosystems and natural habitats that have been degraded or damaged, typically due to human activity or sometimes extreme weather events, some of which may themselves have been amplified by human-driven global warming.

For these subfields where data is scarce by today's Big Data standard, such as Restoration Ecology, techniques such as distribution learning, transfer learning, data augmentation and synthetic data can play a crucial role when it comes to maximizing the knowledge that can be gleaned from whatever data is available. In the present Chapter, we will give examples of how Generative AI has been used in Restoration Ecology in connection with all four of the above aspects, but first, we need to introduce the kind of questions encountered in Restoration Ecology, with a particular view on plant interactions and biodiversity.

2 Biodiversity in Restoration Ecology

One main focus in Restoration Ecology is to restore biodiversity to its former level. Biodiversity describes the variety of life forms within a specific ecosystem or habitat. It is important to note that biodiversity is a complex and multifaceted concept, and although several mathematical definitions exist to quantify it, none can be universally deemed optimal for all purposes. In fact, it is now widely acknowledged that ecosystem services are not solely dependent on species numbers but are closely linked to higher diversity in terms of functional traits and phylogenetic variability [20].

In the context of this chapter, it suffices to acknowledge that biodiversity serves as an indicator of a healthy environment and holds direct implications for human well-being, as recognized by the Convention on Biological Diversity. Because of this, restoration efforts nowadays aim to reinstate biodiversity as the foundation for providing ecological services to humans [4, 18].

Regrettably, the rate of biodiversity loss has escalated in recent years, requiring a growing focus on biodiversity recovery in ecological restoration initiatives [7, 11]. To effectively increase biodiversity in restoration projects, informed decisions regarding species selection based on successional trajectories—the temporal evolution of species abundance within a community—are crucial.

However, due to the intricate nature of plant communities, these successional trajectories often remain elusive and can only be approximated through heuristic approaches [7]. In order to elucidate successional trajectories, one must therefore first understand the ecological interactions between plant species, to which we now turn our focus.

3 Plant Interactions

Understanding the coexistence of different species has always been a fundamental issue in Ecology because it plays a crucial role in maintaining biodiversity [9]. Traditionally, ecologists have endeavored to elucidate the underlying structure of biodiversity by using information that might predict species interactions [3]. Specifically, phylogenetic proximity has been employed as a surrogate to forecast species compatibility, based on the premise that closely-related species tend to exhibit similar behaviors due to shared traits [19].

Yet, recent studies have demonstrated that numerous vital factors contributing to species survival, such as tolerance to abiotic conditions, resource requirements, competitive abilities, response to antagonists and mutualists, and dispersal capacity, are inadequately encapsulated by phylogenetic distance alone, whereas they are accurately captured by the spatial co-occurrence pattern [25]. This discrepancy arises because spatial co-occurrence, in and of itself, provides valuable insights into niche preferences and the outcomes of ecological interactions between multiple species.

Another limitation of traditional approaches to investigate coexistence patterns is that they often focus on pairwise interactions or on a limited number of species, therefore lacking realism [31]. This limitation comes about because the number of indirect interactions would grow exponentially with the inclusion of more species, making it impractical to experimentally quantify them all.

In practice, although some plant interactions may occur on a one-to-one basis, they are frequently influenced by interactions involving three, four, or even more plants or species [27, 31]. Consequently, unraveling such intricate indirect interactions is challenging yet crucial for comprehending community structure [24, 26]. In particular, in highly biodiverse communities comprising numerous species, the sheer magnitude of interactions becomes overwhelmingly large.

While this phenomenon precisely contributes to the richness of diverse communities, it poses significant mathematical challenges, rendering the development of analytic models based on heuristic rules impractical. Therefore, alternative methods are currently needed to evaluate coexistence patterns in complex ecological communities where tens, hundreds, or even thousands of species coexist.

While traditional heuristic modeling approaches have struggled to tackle this complexity, Deep Learning (DL) has shown significant success in handling multi-dimensional problems with higher-order interactions. Crucial to these successes is the expressivity of Artificial Neural Networks (ANNs), enabling them to describe highly-complex mathematical functions without relying on analytical expressions [1, 13, 22, 29].

Although DL (apart from computer vision) has only been applied to a limited number of issues in Ecology, it is well-established that ML can uncover intricate relationships within ecological communities [6], and we now turn to a specific application of GenAI in plant interactions.

4 Distribution Learning for Vegetation Patches

Recognizing the challenges associated with the aforementioned approaches to describing complex interactions in plant communities, our previous work [14] proposed to use GenAI models to extract the underlying rules governing these interactions and to employ the obtained results to predict successional trajectories.

To start with, we focused on plant communities where some plants act as a nurses under whose canopy other species are more likely to grow. Such secondary species are called recruits, and the interaction between nurse and recruit is called facilitation, which is known to be beneficial for the recruit, but has no negative impact on the nurse.

In such facilitation-driven communities, plants tend to grow together and form vegetation patches (clumps of closely-grouped plants that are physically separated from other clumps) [17]. In these patches, species coexistence is influenced by indirect interactions and relies heavily on the composition of the surrounding neighborhood [8, 24].

To attack this problem, we employed two types of models from the field of GenAI: a Generative Adversarial Network (GAN) and a Variational Auto-Encoder (VAE). GANs excel at producing synthetic data that is realistic, while VAEs excel at building a set of more fundamental “latent” variables in which relationships between data points are expressed in a simpler form, and lets us manipulate the data in novel ways. Because of its particular strength, in the present Chapter, we focus on the VAE, and expose some of the principles of its application to plant communities, some of our results, and some directions for future work and applications.

As a first step, we focus on characterizing the plant species composition within vegetation patches, as it provides valuable insights into the probability of species co-occurrence across these patches. By sampling an extensive number of patches and training a GenAI model on it, one is essentially fitting the whole distribution of species co-occurrence, including all the conditional probabilities that rule the presence of any given species as a function of the presence of others. This is akin to GenAI learning the distribution of pixel intensities that belong to plausible human

faces in a random face generator such as <https://this-person-does-not-exist.com/en>, or generate plausible sentences as in <https://openai.com/blog/chatgpt>.

The situation of interest in our case is akin to the one involving pictures of human faces, albeit in a much simpler version. Namely, we use a binary representation of the vegetation patches, whereby the presence or absence of each species is denoted by the digit one or zero, respectively, see Fig. 1.

Each vegetation patch observed in the field thus becomes a data point, with the data being the species composition of the patch. We focused on the most common species in a given study area to the exclusion of those rare species that seldom appear in patches. Since we wanted to make comparisons between plant communities composed of different species in different locations and climates, we had to set an arbitrary threshold of a given number of species, instead of an absolute threshold

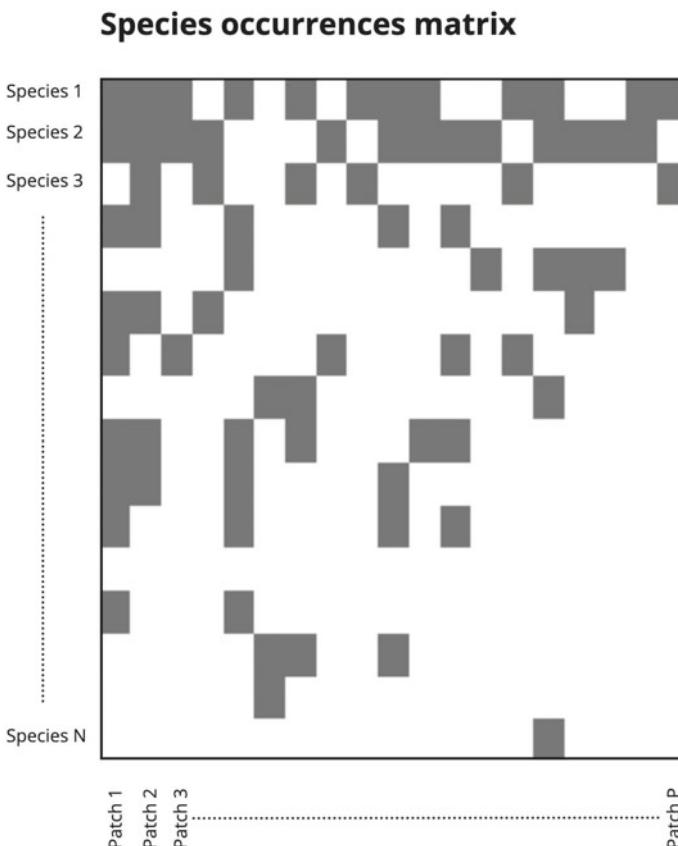


Fig. 1 Representation of a dataset with P patches of vegetation, with the presence of the N most common species in each patch denoted by a dark rectangle (digit 1), and absence of a species as a blank space (digit 0). In this representation, each patch (an entry in the data set) is a list/vector of ones and zeroes, represented here for later convenience as a column of dark/blank pixels

based on species abundances or other. In practice, we focused on the 16 most common species for the datasets we had that involved hundreds or thousands of patches, although it should be possible to apply the method to more species without much loss of performance if enough patches have been sampled.

To study the rules underlying the distribution of this binary representation of patches, we then trained a GenAI model, specifically a VAE [16], to capture the underlying distribution of real data. The VAE comprises two ANNs connected in series. The first ANN takes a real data example as input and encodes it into a set of simplified variables known as the latent space. This encoding process bears similarity to image or audio compression techniques, but incorporates randomness to allow for a smooth behavior in the multi-dimensional latent space. The second ANN, referred to as the decoder, takes a point in the latent space as input and attempts to reconstruct the original data, as in a decompression process, see Fig. 2.

More specifically, the encoder takes inputs from the real data distribution and produces a multivariate Gaussian distribution with a pre-determined number of latent dimensions, see Fig. 2. The decoder then takes a sample point from the latent

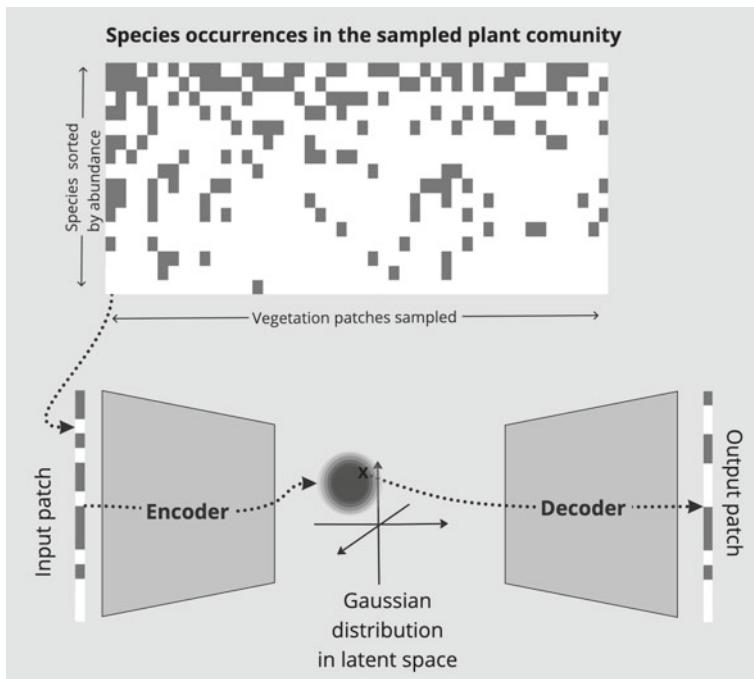


Fig. 2 Representation of the VAE forward pass. A patch is extracted from the dataset and fed to the encoder part of the VAE, yielding a multi-dimensional Gaussian in an abstract (“latent”) space with a selected number of dimensions (“latent variables”). From that Gaussian distribution, a point is sampled (marked by a cross in the Figure), and its coordinates fed to the decoder part of the VAE, yielding an output patch that is similar, but not necessarily identical to the input patch

space and generates simulated data. The intention being that the simulated data point (or vegetation patch in our case) resembles the input, which can be achieved after training.

During the training process of the VAE, the model parameters are adjusted to minimize a loss function. In our implementation, we employed a standard VAE loss function that is the sum of two terms. The first term is the Kullback–Leibler (KL) divergence in the latent space between the prior distribution (a multidimensional unit Gaussian) and the distribution of the input in the latent space [2]. This KL contribution acts as a regularizer for the VAE. The second term in the loss is the reconstruction error, which quantifies the dissimilarity between the input and the final output. In the case of our binary classification task, we used the standard binary cross-entropy as the reconstruction error.

By incorporating these components into the training process, our VAE model aims to learn the underlying structure of the data distribution and to generate simulated data that closely matches the input while adhering to the regularization constraints imposed in the latent space.

The structure of the VAE enables an exploration of this latent space, where the structure of the data can take a simpler form, and underlying relations between data points and variables may be more obvious [23]. The stochastic processes in the latent space are designed to regularize the VAE so that it exhibits smooth properties, enabling the generation of samples with varying attributes by navigating the latent space. This provides a convenient entry point to extract information, simplify the model, and engage in experimentation and modification. It is noteworthy that the latent space has been shown to be capable of extracting interpretable and human-readable variables [15].

In our study, we trained a first VAE with the 16 main species found on limestone soils near Petrer in a semi-arid region of Southeastern Spain, see Fig. 3 for a photograph.

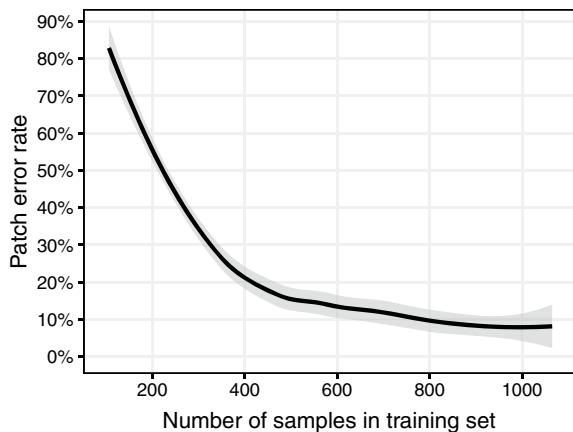
We have found that the VAE could achieve a validation error rate below 10%, meaning that over 90% of the patches are reproduced accurately (and over 99% of the zeroes and ones representing the presence/absence of individual species in the patches), see Fig. 4. In this sense, the VAE was able to learn the distribution of patches for this community, as long as we had a training dataset of several hundred of patches.

As can be seen from Fig. 4, if our dataset had been too small for us to build a training set of at least 500 patches, we would not have been able to train a very useful model in Petrer. In other locations, with different species and interactions, the situation might be different, although for the various plant communities we have worked with, we have found 500 patches to be a good benchmark to train a viable model.



Fig. 3 Vegetation patches near Petrer, Spain

Fig. 4 Validation error rate as a function of the size of the training set, for the plant community in Petrer, Spain



5 Transfer Learning in Community Ecology

What is one to do with smaller datasets? The concept of transfer learning holds great potential in the context of species coexistence, as it allows for the investigation of AI model performance on new datasets, encompassing observations from either a novel site or additional observations within the same site. The original approach [12] involves retraining only the final layers of neurons in an ANN while keeping

the initial layers frozen. This method, requiring a smaller quantity of data, offers a means to expedite the labor-intensive process of collecting field observations.

Furthermore, we expect the similarity between the new site and the reference site to directly influence the amount of new data necessary to achieve a desired level of accuracy on the new site. Notably, instances where transfer learning necessitates only limited amounts of new data would indicate shared assembly rules between the two sites, which the AI model captured comprehensively. An appealing characteristic of an AI model lies in its capacity to transfer knowledge to novel scenarios, enabling ecologists not only to reduce the arduous fieldwork required for training the models but also to extract additional value from past, smaller datasets.

In Ecology, transfer learning has already proven successful in reconstructing the food web of Canadian mammals based on trophic interactions observed in Europe [28]. Despite a mere 4% overlap in species between continents, the phylogenetic information formed a foundation for effective knowledge transfer between the two systems, highlighting the existence of a shared evolutionary pattern shaping the assembly of both trophic networks.

The technique of transfer learning usually comes to the rescue when one has a model trained on a larger dataset for a similar task: that model can be repurposed for the smaller dataset, even if the task is not exactly identical. Taking an example from Computer Vision, for instance object detection, an ANN that classifies cats and dogs could potentially be tested for classification of lynxes and foxes. Its performance is unlikely to be as good in that case, but it could be a good start. In that case, the classes on both datasets have to be matched in order for us to perform the transfer, and it is natural to assume that “cat” should be translated to “lynx” while “dog” translates to “fox”.

The problem is similar in our case: we have to decide how to translate each plant species we found in the large dataset into a plant species present in the smaller dataset. As we just did for the cat and lynx, the natural assumption is to find the most closely related species genetically, and associate this one.

We tested the transferability of our VAE originally trained on the Petrer plant community in Spain [14], to two distinct communities—one in close proximity with a similar Mediterranean climate (southeastern Spain) and another situated far away with a tropical climate (central Mexico). While these communities share the organization of vegetation in spatially delimited patches, they differ in terms of species composition and soil type. Figure 5 summarizes the different steps we followed in order to transfer to a new location the knowledge embedded in a model trained on Petrer.

It turns out that directly transferring without further training works well between two semi-arid communities in Spain, but not from semi-arid Spain to tropical Mexico, see Fig. 6. Also, note that direct transfer using genetic relatedness (i.e. phylogenetic information) yielded worse results than directly training on the small dataset (this will depend on dataset sizes) when transferring from Spain to Mexico rather than from Spain to Spain.

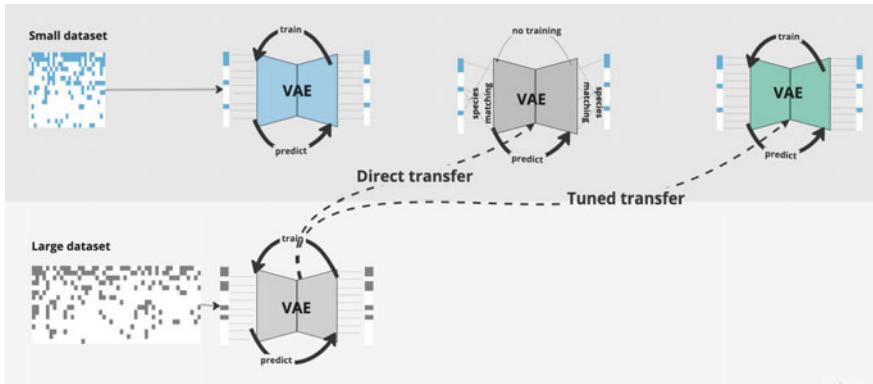


Fig. 5 Conceptual flowchart depicting the various steps taken in the study, and the relations between them. The reference community is the one with the largest dataset, while the target community has fewer observations. A VAE can be trained on a small dataset, usually with limited accuracy. One can also train a VAE on a large dataset from another plant community, and apply it directly to the target dataset, without further training (direct transfer). However, the success of this approach depends on designing a matching procedure between species in both communities, i.e. between rows in the barcodes of both datasets: we tested matching by phylogeny and by species abundance. One can also take the model that has already been trained on the large dataset, and further train it on the target dataset (tuned transfer)

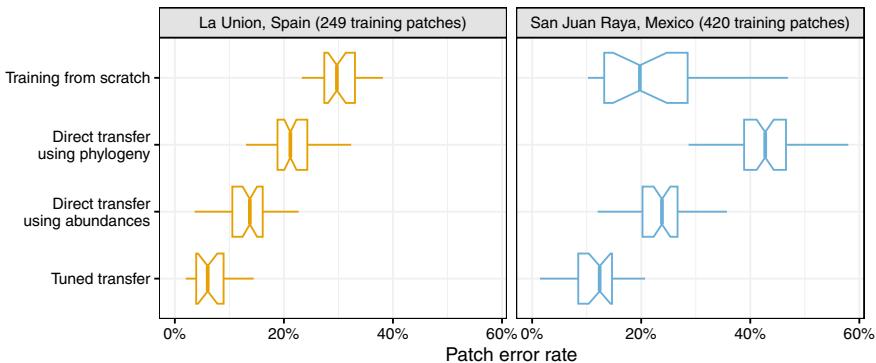


Fig. 6 Error rates for multiple models on two plant communities: a semi-arid one in La Unión, Spain on the left, about 100 km from Petrer, and a tropical one in Mexico, over 9000 km away from Petrer. **a** The top row shows models trained from scratch, which do quite badly in La Unión, due to the small size of the dataset. **b** The second row shows direct transfer after matching species using genetic information. The transfer works well between two semi-arid communities in Spain, but not from semi-arid Spain to tropical Mexico. **c** The third row shows that the transfer using species abundance actually works better in both cases. **d** The bottom row shows that tuning the model after transferring works even better

For both new communities, direct transfer after matching species between communities by order of abundance worked better than using a purely genetic matching (Fig. 6).

Finally, we have compared another way of transferring the models, taken from the concept of fine-tuning in Computer Vision, where a model trained on a large dataset is further trained (or at least some of its layers) on the smaller dataset. Here we actually trained all the layers, with the same original learning rate, and called this procedure “tuned transfer” in order to distinguish it from the usual fine-tuning. For both target communities, this tuned transfer lowers the error rate even further, see Fig. 6.

Although it may not seem surprising that a model trained in semi-arid Spain can be successfully transferred to another plant community 100 km away, it is important to note that the soils are very different: we are transferring from rocky limestone to carbon-poor but metal-rich mine tailings. Even more surprising perhaps, there is not a single species common between the 16 species in one location and the 16 species in the other location.

Perhaps even more surprising is the fact that the VAEs trained on plant communities in semi-arid Spain can do well on another community in tropical Mexico, nearly as well as models trained on the Mexico plant community itself(note that this will of course depend on the size of the target dataset).

This does show that there are some universal laws guiding species interactions (which species lives with which), which are captured by the VAE. As one might expect, these laws seem to apply better to another nearby plant community than to one with very different species and climate. Also, with a very different community, the genetic proximity of species happens to not be a very useful guide to transfer these laws.

Finally, we see that further training on local data of a model pre-trained on a large dataset yields an even better result, as it combines the initial parameters for the NN that are viable even when transferred, with insight from the small set of local data.

6 Data Augmentation for Plant Communities

Data augmentation in Computer Vision consists in modifying images in ways that are expected to preserve the underlying information, or ground truth. For instance, most of us can recognize people or some dog breeds in pictures even if the images are presented with false colors or are in some way cropped, blurred, rotated or flipped.

While it would seem natural to use the VAE precisely to generate new examples of plausible patches, at this stage, we are still more interested in making the models more robust. In the next section, we will talk about using the VAE to generate new, synthetic patches.

Let us explore another interpretation of data augmentation, not with synthetic data, but rather by combining datasets. If we use the example of languages, assuming we are interested in learning the general rules that might be common between languages,

we may want to train a model on a dataset that combines two languages, so that the patterns it extracts are general instead of being language-specific. As a benefit, our model will generalize better to a third language.

The question would then be: how do we match words between languages? In the language context, the logical analogue of using the closest phylogenetic match would be to use the translation of the word. But that does require having a dictionary between the two languages. Another option would be to simply match the words by frequency between the two languages.

Going back to the results of the previous section, we found that instead of matching plant species from dataset to dataset by using a criterion based on phylogenetic distance, we obtained better results by simply matching the species in order of abundance. We use this simpler approach of matching species by frequency as it enables us to automatically match species across datasets, thereby allowing us to grow our dataset.

Another salient point was that further tuning on a small dataset of a pre-trained model allowed to get even better results. This opens the door to a way of improving our models: combining datasets from various plant communities by simply joining them (after matching species by order of abundance), see Fig. 7. Once such a meta-dataset is built from various plant communities, one can then train a single model on all that data. Such a model should generalize better to new communities, instead of specializing to a single one.

In this sense, this is an example of data augmentation, where we extend our dataset by adding to it other datasets that we believe share the similar underlying rules. A model trained on such a meta-dataset should precisely extract the general rules that govern species interactions for all the communities included, and from the results above, we know that such common rules do exist.

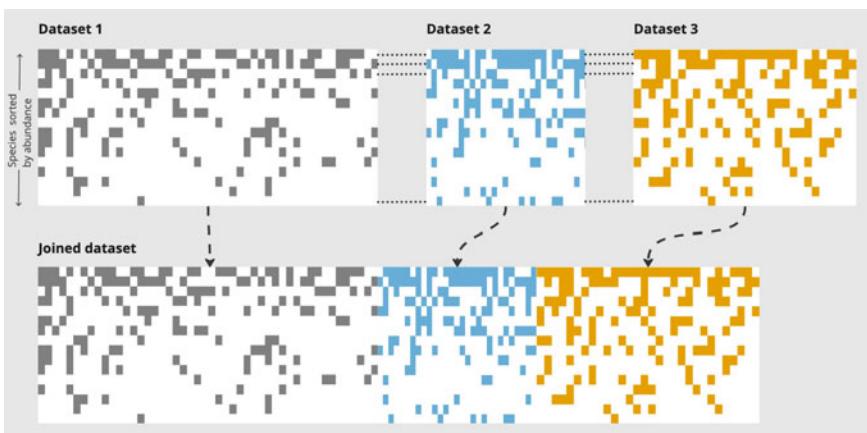


Fig. 7 Joining datasets from different locations, including different species and climate: the datasets are matched by ordering all of them by species abundance, and then joined together to create a larger meta-dataset

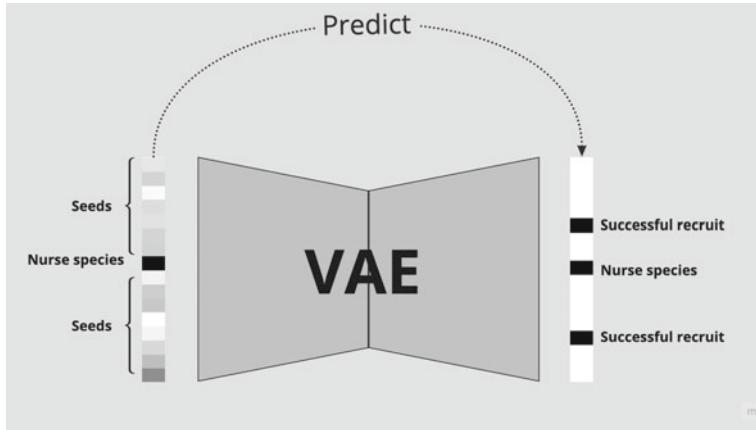


Fig. 8 Using the VAE to predict which seeds actually recruit under the canopy of a given nurse

7 Synthetic Data for Successional Trajectories

Coming back to our motivation of Restoration Ecology for these studies, our initial idea was to predict succession trajectories for restoration efforts in which vegetation is planted in a degraded environment.

A newly planted species on its own can be represented as a patch with a single 1 for the planted (pioneer) species, and zeroes for all other species. But to take into account the possibility of other species' entering the area, we use input patches with a single species turned on, as well as some random small seeds for the others, see Fig. 8. In a sense, the small values act as seeds for these other species, and we are asking the GenAI model which seeds are likely to prosper/recruit under the canopy of a given nurse.

If it is true that the VAE has learned the rules of coexistence of the various species in the community, which can be expressed as the conditional probabilities of finding a species given the presence of other species, then one would expect that it can also predict these recruitment patterns. In particular, we can start looking at various nurses, and ask for which nurse the succession trajectories lead to more biodiverse patches.

8 Summary

We have presented an application of Generative AI (specifically Variational Auto-Encoders) to Ecology, specifically to learn the distribution of species coexistence patterns in plant communities, starting with a dataset from Southeastern Spain.

The accuracy achieved by the aforementioned method requires extensive field data collection, thus impeding its application in situations where sampling proves costly and time-consuming. Yet, including the invaluable information contained within small datasets, replete with spatial and temporal details, could prove crucial in order to test models and broaden their reach so they can be applied to generic situations instead of single communities.

Emerging AI methodologies present a promising avenue to extract and harness information from such modest datasets, thereby unraveling the assembly processes of specific communities and facilitating our understanding of analogous communities at large [30]. Particularly, transfer-learning techniques have been purposefully developed to capitalize on the knowledge gained from training a model on a large dataset, effectively repurposing this knowledge to address similar problems with more limited datasets. By leveraging transfer learning we can unlock the potential of small datasets, enabling a comprehensive exploration of community assembly processes and fostering cross-community insights.

To test knowledge transfer, we used information pertaining to species abundance and phylogenetic relatedness. Our findings demonstrate the successful transfer of information from the community with the largest dataset to communities with fewer data points, substantially improving our ability to predict the coexistence patterns of species within the vegetation patches.

Furthermore, we have used the VAE to predict new patches from input patches containing a single full-grown nurse species, but included small perturbations as placeholders for seeds of recruit species. The idea is that the predicted patches could represent future generations of plants facilitated by the initial pioneer species. Leveraging this capability, one should be able to forecast successional trajectories, which can then be used to guide restoration projects in order to maximize biodiversity.

Overall, we have seen that a VAE offers remarkable results in reproducing observed vegetation patches and even generating previously unsampled patches. Moreover, the VAE's ability to capture successional dynamics can provide valuable insights for guiding restoration efforts.

References

1. Balamurugan, S. A. A., Chitra, P. K. A., & Geetha, S. (2019). Multi label learning approaches for multi species avifaunal occurrence modelling: A case study of south eastern Tamil Nadu. *International Journal of Business Intelligence and Data Mining*, 15, 449–477.
2. Battey, C. J., Coffing, G. C., & Kern, A. D. (2021). Visualizing population structure with variational autoencoders. *G3*, 11(1), jkaa036.
3. Bascompte, J. (2009). Disentangling the web of life. *Science*, 325, 416.
4. Benayas, J. M. R., Newton, A. C., Diaz, A., & Bullock, J. M. (2009). Enhancement of biodiversity and ecosystem services by ecological restoration: A meta-analysis. *Science*, 325, 1121–1124.
5. Borowiec, M. L., Dikow, R. B., Frandsen, P. B., McKeeken, A., Valentini, G., & White, A. E. (2022). Deep learning as a tool for ecology and evolution. *Methods in Ecology and Evolution*, 13, 1640–1660.

6. Bzdok, D., Altman, N., & Krzywinski, M. (2018). Statistics versus machine learning. *Nature Methods*, 15, 233–234.
7. Cardinale, B. J., Duffy, J. E., Gonzalez, A., Hooper, D. U., Perrings, C., Venail, P., Narwani, A., Mace, G. M., Tilman, D., Wardle, D. A., & Kinzig, A. P. (2012). Biodiversity loss and its impact on humanity. *Nature*, 486(7401), 59–67.
8. Castillo, J. P., Verdú, M., & Valiente-Banuet, A. (2010). Neighborhood phylogenetic diversity affects plant performance. *Ecology*, 91, 3656–3663. <https://doi.org/10.1890/10-0720.1>
9. Chesson, P. (2000). Mechanisms of maintenance of species diversity. *Annual Review of Ecology and Systematics*, 31, 343–366.
10. Christin, S., Hervet, É., & Lecomte, N. (2019). Applications for deep learning in ecology. *Methods in Ecology and Evolution*, 10, 1632–1644.
11. Dornelas, M., Gotelli, N. J., McGill, B., Shimadzu, H., Moyes, F., Sievers, C., & Magurran, A. E. (2014). Assemblage time series reveal biodiversity change but not systematic loss. *Science*, 344, 296–299.
12. Frégier, Y., & Gouray, J. B. (2021). Mind2Mind: Transfer learning for GANs. In *International Conference on Geometric Science of Information* (pp. 851–859). Springer, Cham.
13. Harris, D. J. (2015). Generating realistic assemblages with a joint species distribution model. *Methods in Ecology and Evolution*, 6, 465–473.
14. Hirn, J., García, J. E., Montesinos-Navarro, A., Sanchez-Martín, R., Sanz, V., & Verdú, M. (2022). A deep generative artificial intelligence system to predict species coexistence patterns. *Methods in Ecology and Evolution*, 13, 1052–1061.
15. Iten, R., Metger, T., Wilming, H., Del Rio, L., & Renner, R. (2020). Discovering physical concepts with neural networks. *Physical Review Letters*, 124, 010508.
16. Kingma, D. P., & Welling, M. (2013). *Auto-encoding variational Bayes*. arXiv preprint [arXiv:1312.6114](https://arxiv.org/abs/1312.6114).
17. Montesinos-Navarro, A., Pérez-Clemente, R. M., Sánchez-Martín, R., Gómez-Cadenas, A., & Verdú, M. (2020). Phylogenetic analysis of secondary metabolites in a plant community provides evidence for trade-offs between biotic and abiotic stress tolerance. *Evolutionary Ecology*, 34, 439–451.
18. Montoya, D., Rogers, L., & Memmott, J. (2012). Emerging perspectives in the restoration of biodiversity-based ecosystem services. *Trends in Ecology & Evolution*, 27, 666–667.
19. Morales-Castilla, I., Matias, M. G., Gravel, D., & Araújo, M. B. (2015). Inferring biotic interactions from proxies. *Trends in Ecology & Evolution*, 30, 347–356.
20. Navarro-Cano, J. A., Verdú, M., & Goberna, M. (2018). Trait-based selection of nurse plants to restore ecosystem functions in mine tailings. *Journal of Applied Ecology*, 55, 1195–1206.
21. Pichler, M., & Hartig, F. (2023). Machine learning and deep learning—a review for ecologists. *Methods in Ecology and Evolution*, 14, 994–1016.
22. Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., & Sohl-Dickstein, J. (2017). On the expressive power of deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning, PMLR*, vol 70, pp 2847–2854.
23. Ruthotto, L., & Haber, E. (2021). *An introduction to deep generative modeling*. arXiv e-prints, [arXiv:2103.05180](https://arxiv.org/abs/2103.05180).
24. Schöb, C., Armas, C., & Pugnaire, F. I. (2013). Direct and indirect interactions co-determine species composition in nurse plant systems. *Oikos*, 122, 1371–1379.
25. Siebert, A., Laughlin, D. C., & Sabatini, F. M. (2023). *You shall know a species by the company it keeps: leveraging co-occurrence data to improve ecological prediction*. bioRxiv, p. 2023.02.15.528518.
26. Simmons, B. I., Cirtwill, A. R., Baker, N. J., Wauchope, H. S., Dicks, L. V., Stouffer, D. B., & Sutherland, W. J. (2019). Motifs in bipartite ecological networks: Uncovering indirect interactions. *Oikos*, 128, 154–170.
27. Strauss, S. Y. (1991). Indirect effects in community ecology: Their definition, study and importance. *Trends in Ecology & Evolution*, 6, 206–210.
28. Strydom, T., Bouskila, S., Banville, F., Barros, C., Caron, D., Farrell, M. J., Fortin, M. J., Hemming, V., Mercier, B., Pollock, L. J., & Runghen, R. (2022.) Food web reconstruction

- through phylogenetic transfer of low-rank network representation. *Methods in Ecology and Evolution*, 13(12), 2838–2849.
- 29. Tang, L., Xue, Y., Chen, D., & Gomes, C. (2018). Multi-entity dependence learning with rich context via conditional variational auto-encoder. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 32, No. 1).
 - 30. Todman, L. C., Bush, A., & Hood, A. S. (2023). ‘Small Data’ for big insights in ecology. *Trends in Ecology & Evolution*. <https://doi.org/10.1016/j.tree.2023.01.015>
 - 31. van Kleunen, M., Dawson, W., Bossdorf, O., & Fischer. (2014). The more the merrier: Multi-species experiments in ecology. *Basic and Applied Ecology*, 15, 1–9.

On the Effect of Loss Function in GAN Based Data Augmentation for Fault Diagnosis of an Industrial Robot



Ziqiang Pu, Chuan Li, and José Valente de Oliveira

Abstract Intelligent fault diagnosis often requires a balanced dataset which is hard to be obtained in industrial equipments, often resulting in an imbalance between data in normal and data in the presence of faults. Data augmentation techniques are among the most promising approaches to mitigate such issue. Generative adversarial networks (GAN) are a type of generative model consisting of a generator module and a discriminator. Through adversarial learning between these modules, the optimised generator can produce synthetic patterns that can be used for data augmentation. We investigate the role of loss function in improving the training efficiency of GAN. We proposed a generalization of both mean square error (MSE GAN) and Wasserstein GAN with gradient penalty (WGAN-GP), referred to as VGAN (the V-matrix based GAN) to mitigate training instability. Also, we investigate the sliced Wasserstein distance (SWD) as the loss function of a cycle consistency generative adversarial network (CycleGAN), referred to as SW-CycleGAN. Both two models are evaluated on an industrial robot data set. Experimental results show that the proposed loss functions outperform other competitive models especially in terms of computational costs.

Keywords Generative adversarial networks · Data augmentation · Fault diagnosis · V-matrix · Sliced Wasserstein distance · Industrial robots · Imbalanced data

Z. Pu (✉) · C. Li

School of Mechanical Engineering, Chongqing Technology and Business University,
400067 Chongqing, China

e-mail: hojikyou19930127@gmail.com

C. Li

e-mail: chuanli@21cn.com

J. V. de Oliveira

Universidade do Algarve, Faro, Portugal
e-mail: j.valente.de.oliveira@tecnico.ulisboa.pt

Center of Intelligent Systems, IDMEC/LAETA, Instituto Superior Técnico,
Universidade de Lisboa, Lisbon, Portugal

1 Introduction

Industrial robots are widely used in the industry as they can be used in a range of tasks such as assembly, painting or welding [1]. However, the transmission system of the robot is prone to faults due to prolonged working periods [2]. Typically, these faults manifest in the connection parts, bearings, gears, or gear shafts. A faulty robot will be less precise, less efficient, less productive, and less secure. Even though, it could be a tricky task to obtain fault data for such precision machinery. Based on these two issues, the monitoring of the robot health condition with limited data sources is of paramount interest.

Fault diagnosis usually uses data acquired from sensors to detect faults and make prediction on the remain useful life of machinery. While data acquired in healthy state are abundant, data acquired from faulty states are scarce and hardly representative of all possible faults [3]. This could lead to poor diagnosis results. Without appropriate data, the data-driven fault diagnosis simply do not have acceptable performance [4]. It is important to consider a method that mitigates such a data shortage.

To mitigate data scarcity one promising strategy is to use generative adversarial networks (GAN) for data augmentation. A GAN is a generative model proposed by Goodfellow et al. [5] in 2014 that learns data distribution from a given random (e.g., Gaussian) distribution and from real world examples. A GAN consists of two adversarial models (neural networks): a generator and a discriminator. The learning process can be view as a min-max game. The generator produces synthetic examples while the discriminator is a binary classifier that tries to decide whether the current input is a real or a synthetic example. The goal of the generator is to deceive the discriminator by producing real-like samples that are indistinguishable from real ones. Both models improve their performance simultaneously up to a Nash equilibrium [6] using gradient-based optimization techniques [7, 8]. In the initial stage of training GAN, the discriminator can easily differentiate the generated data from the generator. As the adversarial training progresses, distinguishing between real and synthetic samples becomes more and difficult, meaning that the generator has managed to capture the underlying distribution of the real samples.

The loss function is recognized as a crucial factor in the efficiency of GANs training [9, 10]. Both the losses of the generator and the discriminator oscillate during adversarial learning. In addition, when the discriminator gets more accurate than the generator, it may be possible that the whole system does not learn as gradients become unavailable for updating the weights. Designing an appropriate loss function can help GANs to obtain faster and stabler convergence.

The loss function of the original GAN resort to the KL divergence [11] to measure the distance between the generated and real distributions. To improve the training process, many works have paid attention to the loss function of GAN, such as WGAN [12] that resorts to the EMD [13] or Wasserstein-1 distance, Bi-GAN [14], GAN using MSE loss [15], cGAN [16], WGAN-GP [17] and AC-GAN [18]. Besides, the least squares GAN [19] was designed to deal with the vanishing gradient problem. This adopts the least squares loss rather than the divergence-based loss of the conventional

GAN. Geometric GANs was proposed in [20] based on the idea of the support vector machine to search for a decision boundary between real and synthetic examples with the hinge loss [21, 22]. Zhu et al. [23] proposed a cycle consistency GAN (CycleGAN, hereafter also referred to as unconditional CycleGAN) that combines two traditional GANs for the unpaired image-to-image transference and that avoids mode collapse. Che et al. [24] introduced several methods of regularizing the loss function, which can improve the training of GAN.

Another issue in GAN design is the stop criterion for selecting the “best” data generator. The naif selection of the generator obtained in the last epoch might simply not produce the desired results due to the above mentioned oscillation effect. In other words, the desired generator may have appeared in an earlier iteration during the training, not necessarily in the last one. Therefore, keeping the current best generator is a crucial aspect during GAN training.

This chapter addresses both of the mentioned loss function and metric issues in GANs as follows:

- (i) For mitigating oscillations during GAN training, motivated by both the theoretical background and the empirical evidence obtained in classification and regression problems, a V-matrix based regularization is used within the cGAN framework. The V-matrix based criterion proposed by Vapnik et al. [25, 26] generalizes the well-known and widely used MSE criterion. In the same vein, our proposed GAN framework, VGAN, generalizes both the MSE GAN [15] and the WGAN-GP frameworks [27];
- (ii) The sliced Wasserstein distance (SWD) is based on random projections [28] to quantify the two distributions difference with a low-level computational cost comparatively to the Wasserstein distance (WD). SWD has previously been applied in vanilla GAN [29] and in auto-encoders [30]. Motivated by the obtained results in those works and by the need to avoid mode collapse, in this chapter, we investigate the usage of SWD in the loss function of CycleGANs. This distance is first applied to unconditional CycleGANs and then extended to conditional CycleGANs aiming at smoother, faster, more efficient convergence while addressing model collapse [31];
- (iii) A novel early stopping like a strategy that keeps track during training of the most suitable model (Sect. 2.3.2); Besides, another early stopping called model compatibility (MC) rate was adopted in GAN training;
- (iv) The application of the proposed VGAN and Sw-CycleGAN to an industrial robot fault diagnosis where the VGAN is used as a data augmentation tool to cope with an imbalanced data set (Sect. 4) and the SW-CycleGAN is used to transfer healthy states to different fault states also for addressing the imbalanced data problem.

The remainder of this chapter is organized as follows. The theoretical background of GAN is presented in Sect. 2, including conditional WGAN, VGAN, SW-cycleGAN, novel early stoppings. The experimental settings such as neural network architecture, hyper-parameter settings and considered scenarios in both VGAN and SW-cycleGAN are detailed in Sect. 3. Results and discussion are analyzed in Sect. 4. Finally, conclusions is addressed in Sect. 5.

2 Methodology

2.1 On GAN

Both the generator G and the discriminator D forming a GAN are simultaneously refined through an adversarial learning process. G and D play a min-max game where the goal of G is to produce generated samples similar to real samples, and the goal of D is to discriminate the samples generated by G and samples from the real data distribution.

In order to learn the G 's distribution p_g over data x , a prior random distribution (say Gaussian distribution) is defined as p_z and z is a set of noise vectors. Then, G maps from noise p_z to data space p_{data} . $G(z, \theta_g)$ is a neural network with weights θ_g . The $D(x, \theta_d)$ is defined with weights θ_d and will define whether the output of $D(x)$ is from p_g or p_{data} . The discriminator D outputs the probability that x is from the data distribution rather than the generator G . The discriminator D is trained to maximize the probability of giving the correct label to both samples and generated samples. The generator G aims to minimize $\log(1 - D(G(z)))$ (See Fig. 1).

2.1.1 Loss Function

The loss function of GAN is considered as

$$\begin{aligned} \min_{\theta_g} \max_{\theta_d} V(G, D) = & \min_{\theta_g} \max_{\theta_d} E_{x \sim p_{data}} [\log D(x, \theta_d)] \\ & + E_{z \sim p_z} [\log(1 - D(G(z, \theta_g)))] \end{aligned} \quad (1)$$

where $V(G, D)$ is a binary cross entropy loss function, max denotes the maximization of the distribution of the discriminator, min stands for the minimization of the probability in the generator. The gradient-based optimization is used for updating θ_g and θ_d that is solved via the following two gradient updates.

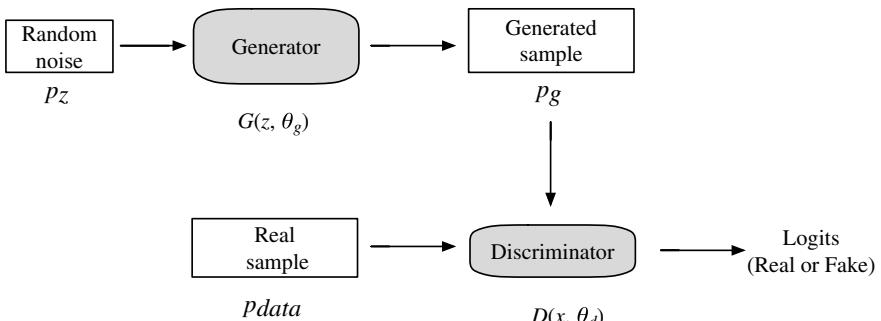


Fig. 1 The architecture of a generative adversarial network (GAN)

$$\theta_g^{t+1} = \theta_g^t + \lambda^t \nabla_{\theta_g} V(D, G) \quad (2)$$

$$\theta_d^{t+1} = \theta_d^t + \lambda^t \nabla_{\theta_d} V(D, G) \quad (3)$$

where θ_g and θ_d are the weights of G and D , respectively. λ is the learning rate, and t denotes the iteration.

2.1.2 Optimization Strategy

Gradient-based optimizations such as adaptive moment estimation (Adam), Root Mean Squared Propagation or stochastic gradient descent (SGD) are widely applied in neural networks to find the minimum of the performance index. The discriminator D and the generator G are trying to minimize their own loss function. For the discriminator D , the loss function is as follows

$$loss_D = \max_{\theta_d} E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p_z} [\log(1 - D(G(z)))] \quad (4)$$

And the loss function of the generator G is given by

$$loss_G = \min_{\theta_g} E_{z \sim p_z} [-\log(D(G(z)))] \quad (5)$$

The min-max game denotes the solution includes minimization and maximization, therefore,

$$loss_D + loss_G = 0 \quad (6)$$

As the loss function of the discriminator, $loss_D$ can be seen as

$$loss_D(\theta_d, \theta_g) = -\frac{1}{2} \int p_{data}(x) [\log D(x)] dx - \frac{1}{2} \int p_g(x) [\log(1 - D(x))] dx \quad (7)$$

Equation (7) represents the classical cross-entropy optimization minimized during the training of a binary classifier with a *sigmoid* output. In the min-max game, the generator G attempts to fool the discriminator by minimizing and maximizing the log probability of the discriminator D . Suppose $y = 0$ means generated data and $y = 1$ denotes real one, the density ratio between the real sample and generated sample in GAN is represented as follows.

$$D(x) = \frac{p_{data}(x)}{p_g(x)} = \frac{p(x|y=1)}{p(x|y=0)} = \frac{D^*(x)}{1 - D^*(x)}, \quad (8)$$

Through training D , it learns to distinguish samples from data for any given G . Combined with Eq. (7), the optimal D^* is searched by the derivative

$$\frac{dloss_D(\theta_d, \theta_g)}{dD(x)} = \frac{1}{2} \int p_{data}(x) \frac{1}{D(x)} dx - \frac{1}{2} \int p_g(x) \frac{1}{1-D(x)} dx, \quad (9)$$

From the necessary condition $\frac{dloss_D(\theta_d, \theta_g)}{dD(x)} = 0$, we have

$$\frac{1}{2} \int p_{data}(x) \frac{1}{D(x)} dx = \frac{1}{2} \int p_g(x) \frac{1}{1-D(x)} dx, \quad (10)$$

Or

$$p_{data}(x) \frac{1}{D(x)} = p_g(x) \frac{1}{1-D(x)} dx, \quad (11)$$

And it can be re-written as

$$D^* = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}. \quad (12)$$

After sufficient training iterations, G and D will converge to $p_g=p_{data}$. This denotes that the range of $D^*(x)$ is between 0 to $\frac{1}{2}$. Next, let $D^*(x)=\frac{1}{2}$ in (1) and we have

$$\max_{\theta_d} V(G, D) = -2 \log 2 + 2(D_{JS}(p_{data}||p_g)), \quad (13)$$

where JS denotes the Jensen-Shannon (JS) divergence. Suppose there are two distributions p and q , the JS can be defined as

$$JS(p||q) = \frac{1}{2}KL(p||\frac{p+q}{2}) + \frac{1}{2}KL(q||\frac{p+q}{2}), \quad (14)$$

where KL is the KullbackLeibler (KL) divergence which can be shown as

$$KL(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx. \quad (15)$$

However, finding the Nash equilibrium in GAN is very challenging as loss functions are non-convex, parameters are continuous, and the parameter space is high-dimensional [9], e.g., an update to θ_d that reduce $Loss_D$ can increase $loss_G$ and vice versa.

2.2 On Wasserstein GAN and Conditional Wasserstein GAN

Since the real data distribution has less overlap with the generated data, JSD-based objective function can be a constant, which causes the vanishing gradient and training oscillation. Arjovsky et al. [12] proposed Wasserstein GAN (WGAN) by using the

Earth-Mover distance (EMD) to replace JSD for measuring the distribution between the real data and the generated one. WGAN resorts to the Kantorovich-Rubinstein duality to define the loss function. WGAN made the progress of the training oscillation in GAN. The loss function of WGAN is shown as follows

$$\begin{aligned} V_{WGAN}(G, D) = & E_{x \sim p_{data}(x)}[D(x)] \\ & + E_{z \sim p_z(z)}[D(G(z))], \end{aligned} \quad (16)$$

where the discriminator D is optimized over the set of 1-Lipschitz functions. Informally, a 1-Lipschitz function is a differentiable function whose gradient has norm at most 1 everywhere. The adoption of such D in (16) allows for a smoother convergence relatively to (1).

However, it can still generate low-quality samples or fail to converge in some settings. In light of that, Gulrajani et al. [17] discovered that the training failures are often due to the use of weight clipping in WGAN to enforce a Lipschitz constraint on the discriminator, which can lead to pathological behaviour. Thus they propose an alternative method for enforcing the 1-Lipschitz constraint instead of clipping weights, penalizing the norm of the gradient of the discriminator concerning its input. Their method (clipping with gradient penalty, also known as WGAN-GP) converges faster and generates higher-quality samples than WGAN. The loss function of WGAN-GP is shown as

$$\begin{aligned} V_{WGAN-GP}(G, D) = & E_{z \sim p_z(z)}[(D(G(z))] \\ & + \lambda_{gp} E_{z \sim p_z(z)}[(\nabla D(\alpha x - (1 - \alpha G(z)))| - 1)^2]. \end{aligned} \quad (17)$$

where α is a user-defined scaling factor and λ_{gp} stands for the gradient penalty coefficient and ∇D is the gradient of D .

Besides, GAN can be extended to a conditional model if the discriminator D and the generator G are conditioned on some extra information y , i.e., labels, shown in Fig. 2. The loss function of conditional GAN (cGAN) is

$$V_{CGAN}(G, D) = E_{x \sim p_{data}(x)}[\log D(x|y)] + E_{z \sim p_z(z)}[\log(1 - D(G(z|y)))] \quad (18)$$

where $D(x|y)$ and $G(z|y)$ represent conditional probabilities while y is the class label information.

Based on this formulation, the inputs are concatenated (embedded) with y to start training. Ultimately, we can generate desired samples by feeding corresponding y . The conditional information y can be formed like class labels [32], or text [33–35]. cGAN has been used for convolutional face generation [36], face editing [37], image translation [38], natural image description [39], 3D-aware scene manipulation [40] and machine translation [41].

Typically, traditional GAN with KL-based loss function [Eq. (1)] can not ensure a better performance which WGAN [Eq. (16)] can improve this strongly oscillates. Also, it is learned through unsupervised learning and its generator generates data for

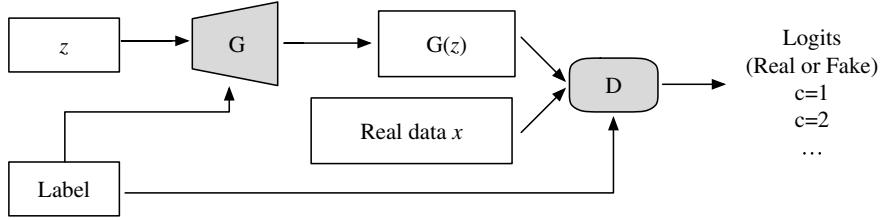


Fig. 2 The architecture of conditional generative adversarial network (cGAN)

a single distribution. If examples are required from k different distributions k GANs are need to be trained; a quite inefficient and time consuming task.

The conditional version of GAN, cGAN (Eq. 18), is designed for giving some controls over the generation process [16]. It can simplify this task as it uses only one model for data generation even from different distributions. This is accomplished by resorting to the label information of the real-world examples. By using this information the learning becomes a supervised learning process. Due to this, several improvements can be conceived for the learning process. One possibility is to adopt a WGAN-GP type of loss function (see Eq. 17) in which case (18) becomes:

$$\begin{aligned} V_{\text{cWGAN-GP}}(G, D) = & E_{x \sim P_{\text{data}}} [D(x|y)] E_{z \sim P_G} [1 - D(G(z|y)|y)] \\ & + \lambda_{gp} E_{x \sim P_{\text{data}}} [(||\nabla D(x|y)|| - 1)^2] \end{aligned} \quad (19)$$

2.3 On VGAN

2.3.1 The Loss Function of VGAN

In the mitigation of oscillations during adversarial training, the loss function plays a critical role in GAN. To address this problem, the previous WGAN (Sect. 2.2) used Wasserstein distance as the loss function for better data generation. But this measurement still has some drawbacks in generating desired signals that need to be further improved. To address this problem, we are proposing the adoption of a V-matrix based regularization criterion within the conditional WGAN framework that considered the mutual point for data generation. The V-matrix based criterion proposed by Vapnik et al. [25, 26] generalizes the well-known and widely used MSE criterion. In the same vein, our proposed GAN framework, VGAN, generalizes both the MSE GAN [13] and the WGAN-GP frameworks. This criterion can be viewed as a generalization of the widely used mean square error criterion. For brevity the idea is presented below for the simplest case. For a more general treatment see [26]. The binary classification problem can be viewed as the problem of estimating the conditional probability of class $y = 1$ given an observation $\mathbf{x} \in \mathbb{R}^d$, $p(y = 1|\mathbf{x})$. Following [26], we can rethink the computation of $f(\mathbf{x}) = p(y = 1|\mathbf{x})$ by viewing it as a solution of the Fredholm integral equation [42]:

$$\int_{\mathbb{R}^d} \theta(\mathbf{x} - \mathbf{x}') f(\mathbf{x}') d\mathbf{x}' = p(y = 1, \mathbf{x}) \quad (20)$$

where the kernel $\theta(z) = 1$ if $z \geq 0$; 0 otherwise. Both the unknown cumulative distribution functions $p(\mathbf{x})$ and $p(y, \mathbf{x})$ can be estimated from iid data $\{\mathbf{x}_i, y_i\}_1^N$ sampled from $p(y, \mathbf{x})$ with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$. In particular, the empirical estimates can be given respectively by

$$\hat{p}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N \theta(\mathbf{x} - \mathbf{x}_i) \quad (21)$$

$$\hat{p}(y = 1, \mathbf{x}) = \frac{1}{N} \sum_{i=1}^N y_i \theta(\mathbf{x} - \mathbf{x}_i) \quad (22)$$

Using these estimates and from (20) one has:

$$\sum_{i=1}^N \theta(\mathbf{x} - \mathbf{x}_i) f(\mathbf{x}_i) = \sum_{i=1}^N y_i \theta(\mathbf{x} - \mathbf{x}_i) \quad (23)$$

Solutions for the above equation can be found by minimizing

$$\rho^2 = \rho(f)^2 \left(\sum_{i=1}^N \theta(\mathbf{x} - \mathbf{x}_i) f(\mathbf{x}_i), \sum_{i=1}^N y_i \theta(\mathbf{x} - \mathbf{x}_i) \right) \quad (24)$$

that, when ρ is viewed as the Euclidean distance, translates into

$$\begin{aligned} \rho^2 &= \int \left(\sum_{i=1}^N \theta(\mathbf{x} - \mathbf{x}_i) f(\mathbf{x}_i) - \sum_{i=1}^N y_i \theta(\mathbf{x} - \mathbf{x}_i) \right)^2 d\mathbf{x} \\ &= \sum_{i=1}^N \sum_{j=1}^N f(\mathbf{x}_i) f(\mathbf{x}_j) V(i, j) \\ &\quad + \sum_{i=1}^N \sum_{j=1}^N y_i y_j V(i, j) - 2 \sum_{i=1}^N \sum_{j=1}^N f(\mathbf{x}_i) y_j V(i, j) \end{aligned} \quad (25)$$

where $V = [V(i, j)]$ is the so-called V-matrix and

$$V(i, j) = \int \theta(\mathbf{x} - \mathbf{x}_i) \theta(\mathbf{x} - \mathbf{x}_j) d\mathbf{x} \quad (26)$$

In the d -dimensional case where $\mathbf{x} \in [0, c]^d$, one has

$$V(i, j) = \int \prod_{k=1}^d \theta(\mathbf{x} - \mathbf{x}_i)\theta(\mathbf{x} - \mathbf{x}_j)d\mathbf{x} \quad (27)$$

Furthermore, Eq. (27) with $\mathbf{c} = (c_1, \dots, c_d)^T$ can be represented as

$$V(i, j) = \prod_{k=1}^d (c_k - \max(x_i^k, x_j^k)) \quad (28)$$

where c_k represents the maximum value of the k -th coordinate ($k = 1, \dots, d$) while x_i^k denotes the k -th coordinate of the i -th example ($i = 1, \dots, n$). However, this multiplicative form is hard for high-dimensional space [26]. In this work, the alternative additive form of (28) is used, i.e.,

$$V(i, j) = \sum_{k=1}^d (c_k - \max(x_i^k, x_j^k)) \quad (29)$$

From (25) we have

$$\rho_v^2 = \sum_{i=1}^N \sum_{j=1}^N (y_i - f(\mathbf{x}_i))(y_j - f(\mathbf{x}_j))V(i, j) \quad (30)$$

Notice that (30) holds as a special case the classical mean square error

$$\rho_{\text{MSE}}^2 = \sum_{i=1}^N (y_i - f(\mathbf{x}_i))^2 \quad (31)$$

whenever the V-matrix equals the identity matrix. That is, while in (31) only the residuals $\Delta_i = y_i - f(\mathbf{x}_i)$ are taken into account while searching for f , in (30) both the residuals and the relative dispersion of observations \mathbf{x}_i and \mathbf{x}_j are considered, resulting in a more general criterion.

Now, aiming at smoothing the learning process, motivated by the above development, and by the empirical evidence from some classification and regression problems (e.g., [25, 26]), we regularize (19) using the (30) as regularizer yielding

$$V_{\text{VGAN}}(G, D) = V_{\text{cWGAN-GP}}(G, D) + \gamma \rho_v^2(G) \quad (32)$$

where γ is a user-defined regularizing coefficient. Notice that the regularizer $\rho_v^2(G)$ concerns only the generator G , where relatively to (30) \mathbf{x}_i and y_i are now the generator

i -th input and corresponding desired output, respectively. For brevity, hereafter (32) is referred to as V-regularization, and this type of generative model is VGAN. Notice that VGAN can be viewed as a generalization of the MSE GAN [15] and degenerates into it for $\gamma = 1$ and V-matrix equal to the identity matrix. Also, VGAN can be viewed as a generalization of the cWGAN-GP (19) and degenerates into it for $\gamma = 0$. To the best of our knowledge this is the first study where the V-matrix is applied in the realm of GANs and deep learning in general. For comparison purposes, in Sect. 4, the traditional MSE based regularization, i.e., MSE GAN

$$V_{\text{MSE}}(G, D) = V_{\text{cWGAN-GP}}(G, D) + \gamma \rho_{\text{MSE}}^2(G) \quad (33)$$

is also considered.

2.3.2 The Early Stopping in VGAN

Classification accuracy is a metric that summarizes the performance of a classification model as the proportion of correct predictions divided by the total number of predictions. Inspired by this, we proposed new stop criteria named early stopping, which used the idea of classification accuracy for evaluating the performance of GAN. Roughly speaking, early stopping means interrupting the learning process whenever a theoretical, such as a bias-variance trade-off, or, e.g., a holdout validation-based criterion is verified.

We follow a holdout strategy where the available real data is divided into disjoint training and validation sets. In the following, these are denoted as $\{\mathbf{x}^{(t)}, \mathbf{y}^{(t)}\}$ and $\{\mathbf{x}^{(v)}, \mathbf{y}^{(v)}\}$, respectively, the superscripts (t) and (v) referring to the training and validation sets, respectively. The input data of the training set is $\mathbf{x}^{(t)} = \{x_1^{(t)}, x_2^{(t)}, \dots, x_N^{(t)}\}$ while the label information is given by $\mathbf{y}^{(t)} = \{y_1^{(t)}, y_2^{(t)}, \dots, y_N^{(t)}\}$. A similar notation is used for the validation set.

Let K be the number of epochs. Thus, K successive generators will be available during learning. The generator available at iteration j , G_j , ($j = 1, 2, \dots, K$) generates $\mathbf{x}_j^{(g)} = \{x_{j1}^{(g)}, \dots, x_{jN}^{(g)}\}$ when $\mathbf{y}^{(t)}$ is presented at its input. Thus, we can write

$$\mathbf{x}_j^{(g)} = G_j(z|\mathbf{y}^{(t)}) = G_j(\mathbf{y}^{(t)}) \quad (34)$$

Notice that the superscript (g) refers to *generated* data; as before, z is a random variable (e.g., Gaussian).

Now consider a classifier, say a random forest R , the formulation of R can be shown as

$$M = \{\hat{n}^*, j^* | I(\hat{m}, \hat{n})\} \quad (35)$$

where I stands for the Gini index, m is the tree node, \hat{n} refers to the splitting node, \hat{n}^* represents the optimal splitting node and j^* is the optimal feature.

For each j of the K generated data sets $\mathbf{x}_j^{(g)}$, train a model M_j , s.t.,

$$M_j = R(\mathbf{x}_j^{(g)}) = R(G_j(\mathbf{y}^{(t)})) \quad (36)$$

At this time, employ the validation set to evaluate the j -th model as follow:

$$y_j^{(p)} = M_j(\mathbf{x}_j^{(v)}) \quad (37)$$

where the superscript (p) refers to a predicted label.

The performance of the j -th model M_j is viewed as a proxy for the performance of the j -th generator G_j . We use accuracy (Acc) for evaluating such performance. The Acc of the j -th model is computed in a straightforward way:

$$Acc(G_j) = \frac{1}{N} \sum_{i=1}^N \mathbb{1}(y_{ji}^{(t)}, y_{ji}^{(p)}) \quad (38)$$

where the $\mathbb{1}(\alpha, \beta)$ is the indicator function that returns 1 if α equals β ; 0 otherwise.

We term G_{\max} the generator whose generated data $\mathbf{x}_j^{(g)}$, yields the max accuracy, i.e.,

$$G_{\max} = \arg \max_{G_j; j=1, \dots, K} Acc(G_j) \quad (39)$$

At this point, one can legitimately ask why using all of this if one can directly use the GAN discriminator to compute the accuracy. The reason is twofold. On one hand, during learning (especially in the initial epochs) the discriminator is not yet properly trained, so it is still a poor option for evaluating generated data; On the other hand, if we would get a poor performance we were not able to tell which of the generator or discriminator would be to blame. So by using a classifier such as random forest for evaluating the generated data we can best control the selection of G_{\max} . Notice that the discriminator is still and indeed an important part of the GAN learning process as only simultaneous learning of the generator and the discriminator allows for the best results [43].

2.4 On SW-CycleGAN

2.4.1 The Sliced Wasserstein Distance

The Wasserstein- p distance between distributions P_d and $G_\theta(P_z)$ is given by Freirich et al. [44]:

$$W_p(P_d, G_\theta(P_z)) = \inf_{\gamma \in \Pi(P_d, G_\theta(P_z))} (\mathbf{e}_{(x,y) \sim \gamma} [|x - y|^p])^{\frac{1}{p}}, \quad (40)$$

where for p is a user-defined positive integer, x denotes real data points, y represents synthetic one from $G_\theta(P_z)$, $\Pi(P_d, G_\theta(P_z))$ stands for the set of all joint distributions

$\gamma(x, y)$. Informally, $\gamma(x, y)$ denotes how much 'mass' must be transposed from x to y in order to transform the distribution P_d into $G_\theta(P_z)$. With Kantorovich-Rubinstein duality, (40) yields

$$\begin{aligned} W_p(P_d, G_\theta(P_z)) &= \sup_{\|f_w\|_L \leq 1} \mathbf{e}_{x \sim P_d}[f_w(x)] \\ &\quad - \mathbf{e}_{z \sim P_z}[f_w(G_\theta(z))], \end{aligned} \quad (41)$$

where the sup is defined over all 1-Lipschitz functions $f_w : \chi \rightarrow \mathbb{R}$. Therefore, (41) can be used in (16) in the definition of WGAN.

With this in mind, consider the case $p \neq 1$ in (40). Suppose that data x belongs to the real data set \mathcal{D} and the generated data $\hat{x} = G_\theta(z)$ belongs to a synthetic data set \mathcal{F} (\mathcal{D} and $\mathcal{F} \in$). The minimum distance estimation is defined as [45]:

$$\arg \min_{\theta} |P_d, G_\theta(P_z)|, \quad (42)$$

where $|P_d, G_\theta(P_z)|$ denotes a divergence between probability distributions. Using a Wasserstein measurement, (42) can be re-written as

$$\arg \min_{\theta} W_p, \quad (43)$$

Considering the Wasserstein quadratic distance $W_2^2(\mathcal{D}, \mathcal{F})$ between two distributions, (40) can be re-written as [46]

$$W_2^2(\mathcal{D}, \mathcal{F}) = \frac{1}{|\mathcal{F}|} \min_{s \in \sum_{|\mathcal{F}|}} \sum_{i=1}^{|\mathcal{F}|} \|x_{s_{\mathcal{D}}(i)} - \hat{x}_{s_{\mathcal{F}}(i)}\|_2^2, \quad (44)$$

where $\sum_{|\mathcal{F}|}$ is the set of all projections of $|\mathcal{F}|$ points, $s(i)$ is the index needs to be searched for mapping \hat{x} to x . For facilitating the convergence of (44), the optimal searching for s^* is defined as an integer linear program with constant matrices M , i.e.,

$$W_2^2(\mathcal{D}, \mathcal{F}) = \frac{1}{|\mathcal{F}|} \min_M \sum_{i=1}^{|\mathcal{F}|} \sum_{j=1}^{|\mathcal{D}|} M_{i,j} \|x_{s_{\mathcal{D}}(j)} - \hat{x}_{s_{\mathcal{F}}(i)}\|_2^2, \quad (45)$$

where the matrix M is also known as a permutation matrix that can be estimated through a linear programming solver. For approximation between these two distributions with permutation matrix M , a summation over random directions of vectors Ω in the projection sphere (Fig. 3) is subject to the following sorting conditions [29]:

$$x_{s_{\mathcal{D}}(i)}^r \leq x_{s_{\mathcal{D}}(i+1)}^r, \exists i \in \{0 \leq i < |\mathcal{D}|\}, \quad (46)$$

$$\hat{x}_{s_{\mathcal{F}}(i)}^r \leq \hat{x}_{s_{\mathcal{F}}(i+1)}^r, \exists i \in \{0 \leq i < |\mathcal{F}|\}, \quad (47)$$

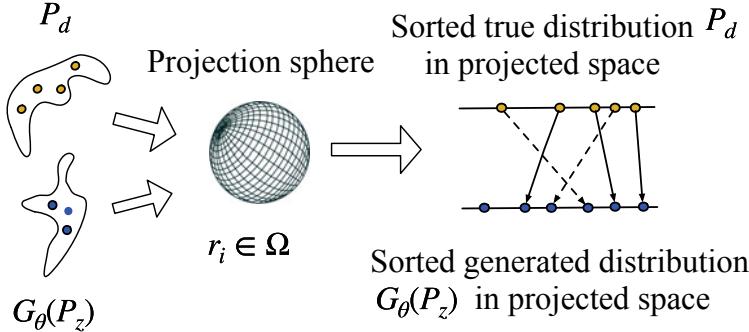


Fig. 3 Random projections and permutation of two distributions P_d and P_z , adapted from [47]

As illustrated in Fig. 3, the data and synthetic points are projected into the one-dimensional spaces by sorting the projections of all possible directions r_i on the projection sphere Ω [46]:

$$SW_2^2(\mathcal{D}, \mathcal{F}) = \oint_{r_i \in \Omega} W_2^2(\mathcal{D}^{r_i}, \mathcal{F}^{r_i}) dr, \quad (48)$$

Therefore, the approximation for sliced Wasserstein distance SWD_2 with all possible directions over Ω can be found by computing the average distance between sorted samples [48]:

$$SWD_2 = \min_{\theta} \frac{1}{|\Omega|} \sum_{r_i \in \Omega} W_2^2(\mathcal{D}^{r_i}, \mathcal{F}^{r_i}), \quad (49)$$

Specifically, combined with (44), the above expression can be re-written as

$$SWD_2 = \min_{\theta} \frac{1}{|\Omega|} \sum_{i=1}^{|\mathcal{F}|} \|x_{s_{\mathcal{D}}(i)}^r - \hat{x}_{s_{\mathcal{F}}(i)}^r\|_2^2. \quad (50)$$

2.4.2 Unconditional Cycle Generative Adversarial Networks

In [23], a new variant of GAN, CycleGAN, was proposed for unpaired image-to-image translation [49]. A CycleGAN is composed of two classic GANs. Since these two GANs have a symmetric architecture, only one part of the CycleGAN is detailed in Fig. 4. For illustrative purposes, consider the MNIST dataset. In the first GAN, real images (say images of ‘3’) from domain A are sent to the generator G12 to produce synthetic images from domain B (say images of ‘0’), which in turn are compared with real domain B images in the discriminator D12. At the same time,

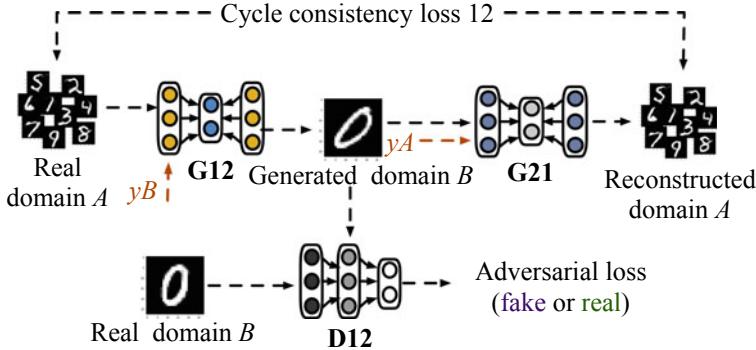


Fig. 4 The flow chart of a conditional CycleGAN

the generated domain B images will be passed through another generator G_{21} to reconstruct images of the domain A . The loss between real domain A images and reconstructed domain A images is the cycle consistency loss 12. The second GAN uses real images from domain B to generate images for domain A via generator G_{12} . Adversarial training will compare the generated domain A images with real domain A images through the discriminator D_{12} .

The involved losses of a CycleGAN are [23]:

$$V_{CycleGAN} = L_{adv}^{12} + L_{adv}^{21} + \lambda_c * L_{cyc}, \quad (51)$$

where L_{adv}^{12} and L_{adv}^{21} are adversarial losses, L_{cyc} is the cycle consistent loss, λ_c is the corresponding hyper-parameter. The expressions for these losses are as follows.

$$\begin{aligned} L_{adv}^{12} &= \mathbb{E}_{x \sim P_B} [D_{12}(x_B)] + \mathbb{E}_{x \sim P_A} [(D_{12}(G_{12}(x_A))] \\ &\quad + \lambda \mathbb{E}_{x \sim P_A} [(|\nabla D_{12}(\alpha_{12}x - (1 - \alpha_{12}G_{12}(z)))| - 1)^2], \end{aligned} \quad (52)$$

$$\begin{aligned} L_{adv}^{21} &= \mathbb{E}_{x \sim P_A} [D_{21}(x_A)] + \mathbb{E}_{x \sim P_B} [(D_{21}(G_{21}(x_B))] \\ &\quad + \lambda \mathbb{E}_{x \sim P_B} [(|\nabla D_{21}(\alpha_{21}x - (1 - \alpha_{21}G_{21}(z)))| - 1)^2], \end{aligned} \quad (53)$$

$$\begin{aligned} L_{cyc} &= \mathbb{E}_{x_A} [|x_A - G_{21}(G_{12}(x_A))|_2] \\ &\quad + \mathbb{E}_{x_B} [|x_B - G_{12}(G_{21}(x_B))|_2]. \end{aligned} \quad (54)$$

This scheme is particularly effective for dealing with mode collapse as it forces each generator to produce a new output for each new input. However, training for multi-domains (classes) is quite time-consuming as it needs to be repeated for each new domain pairs. In the MNIST example, for each and every digit in domain B .

2.4.3 Conditional Variant of Cycle Generative Adversarial Network

A conditional variant of CycleGAN can be considered that adds extra information (a label) to each input which is used to specify the desired output [50]. Returning to the MNIST example as illustrated in Fig. 4, the first GAN receives a domain A image together with a label $yB \in \{1, 2, \dots, 9\}$ at the generator G_{12} to generate the corresponding yB domain image. Generated domain B images will compare with real domain B images in the discriminator D_{12} . At the same time, generated domain B images and labels yA will be sent to the other generator G_{21} to reconstruct domain A images. In the conditional case, the losses are:

$$\begin{aligned} L_{adv}^{12} = & e_{x \sim P_B} [D_{12}(x_B | yB)] \\ & + e_{x \sim P_A} [(D_{12}(G_{12}(x_A | yB) | yB))] \\ & + \lambda e_{x \sim P_A} [(\nabla D_{12}(\alpha_{12}x - (1 - \alpha_{12}G_{12}(x_A | yB))) \\ & - 1 | yB)^2], \end{aligned} \quad (55)$$

$$\begin{aligned} L_{adv}^{21} = & e_{x \sim P_B} [D_{21}(x_A | yA)] \\ & + e_{x \sim P_A} [(D_{21}(G_{21}(x_B | yA) | yA))] \\ & + \lambda e_{x \sim P_A} [(\nabla D_{21}(\alpha_{21}x - (1 - \alpha_{21}G_{21}(x_B | yA))) \\ & - 1 | yA)^2], \end{aligned} \quad (56)$$

$$\begin{aligned} L_{cyc} = & e_{x_A} [| | x_A - G_{21}(G_{12}(x_A | yB) | yA) | |_2] \\ & + e_{x_B} [| | x_B - G_{12}(G_{21}(x_B | yA) | yB) | |_2]. \end{aligned} \quad (57)$$

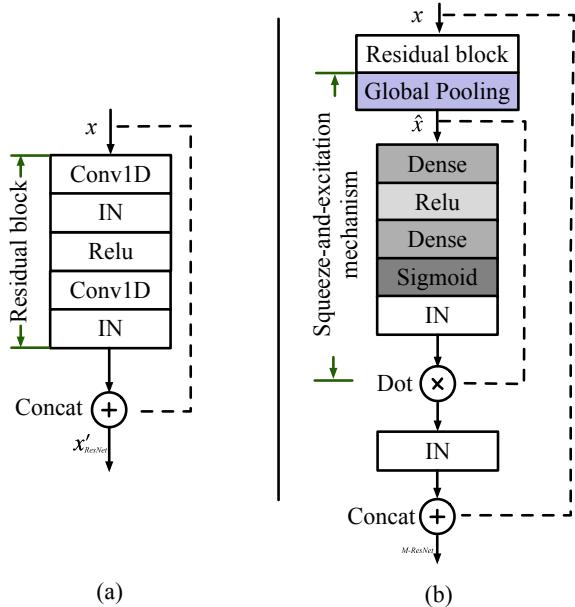
2.4.4 Residual Networks

To address gradient degradation in deep neural networks, He et al. [51] proposed the residual network (ResNet). The ResNet consists of a series of residual blocks, whose basic diagram is shown in Fig. 5a, the corresponding expression being

$$x'_{ResNet} = x + Q(x), \quad (58)$$

where $Q(x)$ is the residual mapping, x is the input, and x'_{ResNet} is the output from the residual block. Even though this process can release the pressure when adding more layers to the neural network, it may cause a certain insensitivity to the inputs. To this end, a modified ResNet network, i.e., ResNet with the squeeze-and-excitation mechanism (SEM) [52], can be considered to enhance the sensitivity to the inputs. SEM consists of the squeeze block and the excitation block. Firstly, the input vectors are $Q(x)$ and then input to the global average pooling (the squeeze block) is used to shrink inputs in the squeeze stage:

Fig. 5 The ResNet architecture: **a** the classical ResNet block; **b** the modified ResNet block



$$\hat{x} = F_{sq}(Q(x)) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W Q(x)(i, j), \quad (59)$$

where F_{sq} is the squeeze operation, \hat{x} is the output from the squeeze block, also known as the input for the excitation block. Then in the excitation block, two dense layers are used to learn the non-linear relationship between each channel, which is shown as

$$s = F_e(\hat{x}, \mathbf{W}) = f_2(W_2 \times f_1(W_1 \times \hat{x})), \quad (60)$$

where F_e is the excitation operation, f_1 and f_2 are Relu and Sigmoid activations, respectively; W_1 and W_2 are weights for these two dense layers; s is the weight for the feature channel. Next, the output of excitation block s and the \hat{x} are combined through the so-called 'dot' operation, i.e.,

$$F_{sem} = F_{scale}(\hat{x} \times s), \quad (61)$$

Finally, with instance normalization of F_{sem} , one gets (see also Fig. 5b):

$$x'_{M-ResNet} = F_{scale}(F_{sem} + x). \quad (62)$$

2.4.5 On the Early Stopping in SW-CycleGAN

Selecting a generator during the adversarial learning is difficult since the convergence curve is not decreasing monotonically, and the generator obtained in the last generator may not be the best one. Therefore, keeping the current best generator is a crucial aspect during GAN training.

The following model compatibility (MC) rate was set as an early stopping in SW-CycleGAN that can be used to access the generator qualitatively:

$$MC_i(\mathcal{D}, \mathcal{F}) = 1 - \frac{e(\mathcal{D}, \mathcal{F})_{i=K}, K \geq 0}{e(\mathcal{D}, \mathcal{F})_{i=0}}, \quad (63)$$

subject to $e_K < e_0$

where i refers to the iteration number, e is the metric measuring how close \mathcal{D} , the real data set, is from \mathcal{F} , the corresponding generated data set. We defined e as follows:

$$e_i = \frac{1}{|D|} \sum_{j=1}^{|D|} \|x_B(j), G_{12}^i(x_A(j))\|_2 + \|x_A(j), G_{21}^i(x_B(j))\|_2. \quad (64)$$

where $\|\cdot\|_2$ as the Euclidean metric. The lower the value of this expression the better, meaning that the generated domain has more overlapped with the real domain.

As for the MC rate, initially, e_0 is large but will be reduced with training. The closer the two data sets \mathcal{D} and \mathcal{F} are, the closer e_i will be to zero, the closer $MC_i(\mathcal{D}, \mathcal{F})$ will be of to 1. The current best generator is the one with the highest MC .

2.4.6 Procedure of the Proposed SW-CycleGAN

The whole procedure of the proposed model is shown in Algorithm 1. Besides, the Algorithm for calculating SWD is specified in Algorithm 2.

3 Experiments

3.1 The Industrial Robot Test Rig and Data Set

The main objective of fault diagnosis is to detect and classify incipient faults while the robot is operating. The test rig is shown in Fig. 6 and consists of an industrial robot (Brtirus 1510A), an accelerometer (PCB 622B01), a NI data acquisition system and a laptop (DELL XPS 9380). The accelerometer is mounted on the robot arm. The robot has six axes labeled from J1 to J6. Each axis has an RV reducer to operate

Algorithm 1 Sliced Wasserstein conditional CycleGAN

input : Mini batch images x_A and y_B in source domain; Mini batch images x_B and y_A in target domain; number of iterations S ;batch size K .

output: G_{best} : The generator yielding max MC rate over the considered number of epochs, K .
Set $MaxMC$ rate= 0 **for** $i = 1$ to S **do**

- for** $j = 1$ to K **do**
 - Calculate adversarial losses (L_{adv}^{12} and L_{adv}^{21}) by (55) and (56) with Sliced Wassrestein distance. (See Algorithm 2)
 - Calculate cycle consistency losses L_{cyc} with (57).
 - Obtain the overall losses for conditional CycleGAN through (51).
 - Calculate the initial L_2 distance with (63).
 - Update and optimize weights $\theta \in \{\theta_{g12}, \theta_{g21}, \theta_{d12}, \theta_{d21}\}$
 - Calculate the updated L_2 distance with (63)
 - Calculate the current MC rate using with (64) **if** $MC > MaxMC$ **then**
 - $| MaxMC = MC \quad G_{best} = G_i$
 - end**
- end**

return G_{best}

Algorithm 2 Sliced Wasserstein distance

input : Images $x \in \{x_A, x_B\}$; Generated images $y \in \{G_{12}(x_A), G_{21}(x_B)\}$; sample size n ;number of random projections m , learning rate α .

output: Optimized weight θ_g

while θ_g not converged **do**

- Sample random projection directions $\Omega = \{r_{1:m}\}$; Inirial loss $L_g=0$.
- for each** $r_i \in \Omega$ **do**
 - $x^r \leftarrow \{r^T x_i\}_{i=1}^n, y^r \leftarrow \{r^T y_i\}_{i=1}^n;$
 - $x_s^r \leftarrow sorted x^r, y_s^r \leftarrow sorted y^r$ with (46) and (47) ;
 - $L_g \leftarrow L_g + \frac{1}{n} ||x_s^r - y_s^r||^2;$
- end**
- return** $\frac{L_g}{m}$
- $\theta \leftarrow \theta - \alpha \nabla_\theta L_g$

end

the robot's dynamic movement. Therefore, the central part of dynamic monitoring for the fault diagnosis is on the RV reducer. During measurements, the working conditions of this industrial robot are with low-speed rotation of 600 r/min and a heavy load of 9.6kg. The robot is moved by the motors, and the teaching box gives the instructions to the robot to start its next movement. At the beginning of the process, the robot is in its original position of 0 °C. Firstly, it will start back and forth movement from –115 to 140 °C of the limit range point in the first axis. Secondly, the same movement and the same limit range which is from –50 to 35 °C. Thirdly, the robot will move from –60 to 90 °C. Fourthly, the same configuration of movement is from –180 to 180 °C. Fifthly, the movement range will be decreased that the range is from –90 to 90 °C. At last, the robot will move from –180 to 180 °C and stop in the original place. This series of dynamic movements is only one experiment

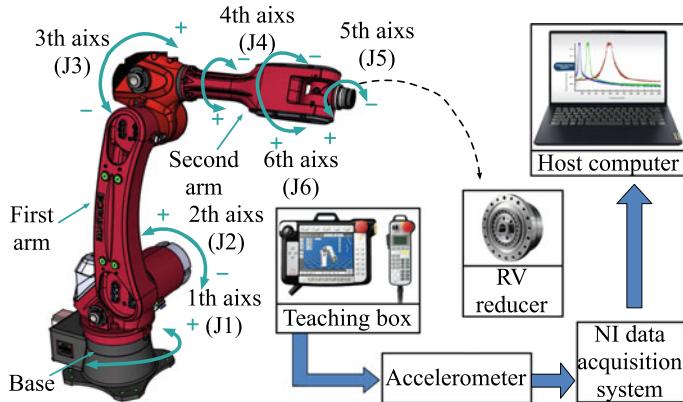


Fig. 6 The experimental apparatus

Table 1 Different fault patterns in the industrial robot

Fault id	Part	Fault type
C_0	None	Healthy
C_1	Sun gear A	Pitting
C_2	Sun gear A	Broken tooth
C_3	Planetary gear B	Cracking
C_4	Planetary gear A	Cracking
C_5	Planetary gear B	Broken tooth
C_6	Sun gear B	Broken tooth
C_7	Sun gear A	Cracking
C_8	Planetary gear A	Broken tooth

process. Next, we replace the faulty part to restart the above movement for the next experiment. Finally, the signal in each channel is collected by the NI acquisition system, which is an analog-to-digital conversion system that the digital samples are collected with an interface on the laptop. Note that all the data are collected in the School of mechanical engineering, Dongguan University of Technology, China.

Table 1 shows the each operating condition and Fig. 7 shows an example of each one of the 8 types of faults. Measurements were performed at a sampling rate of 100 kHz. The sampling duration of each measurement is 20 s. The sampling interval was set to 0.2 s. Thus, 20,000 observations were obtained in each fault condition, and 20 k points were chosen for each observation. For hold-out validation, the data set was divided into two disjoint subsets, the training and the test (sub)sets. The training set has 70% of the data while the test set has the remaining 30%. Figure 8 shows the time length of the vibration signal acquired in each one of the fault types.

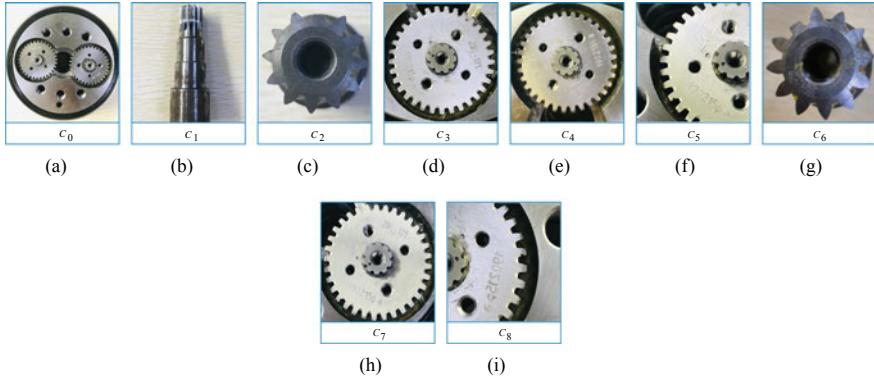


Fig. 7 Examples of each one of the 8 monitoring conditions: **a** Healthy state; **b** Pitting in Sun gear A; **c** Broken tooth in Sun gear A; **d** Cracking in Planetary gear B. **e** Cracking in Planetary gear A; **f** Broken tooth in Planetary gear B; **g** Broken tooth in Sun gear B; and **h** Cracking in Sun gear A; **i** Broken tooth in Planetary gear A

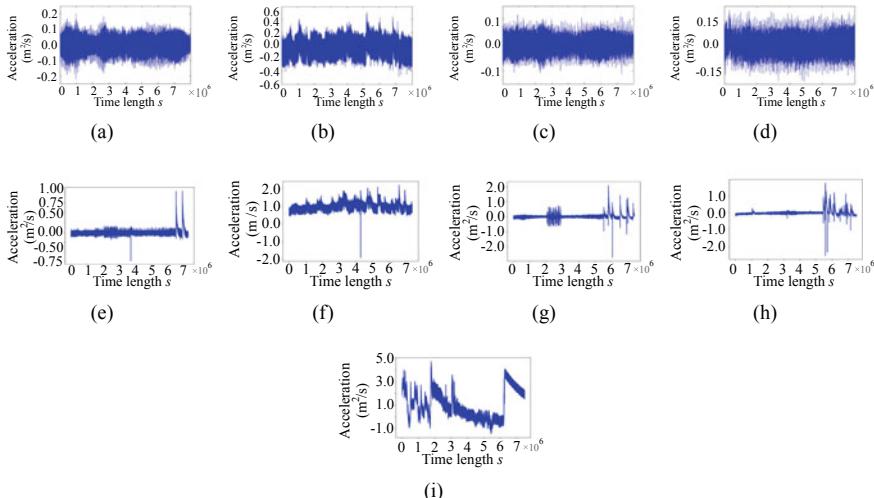


Fig. 8 The time length of vibration signal visualization in each fault condition: **a** Healthy condition (C_0); **b** Pitting in Sun gear A (C_1); **c** Broken tooth in Sun gear A (C_2); **d** Cracking in Planetary gear B (C_3); **e** Cracking in Planetary gear A (C_4); **f** Broken tooth in Planetary gear B (C_5); **g** Broken tooth in Sun gear B (C_6); **h** Cracking in Sun gear A (C_7); **i** Broken tooth in Planetary gear A (C_8)

3.2 Experimental Settings of VGAN

To validate the performance of VGAN, we start from a perspective of the fault diagnosis under the imbalanced data set with the same faults configuration (C_0 , C_1 , C_2 , and C_3). In the training phase, the training data is used together with fault label information in the training of the VGAN. After training the VGAN, G_{\max} (39) is used for generating enough synthetic data for achieving an augmented balanced training set. This is the data set used for training the fault classifier. The validation data set which was never used during training is subsequently used for assessing the fault classification performance.

3.2.1 Neural Network Architectures

In this application, both the discriminator and the generator are multi-layer-perceptrons with a fully connected topology. With the label information, the topology of the generator is 67:1024:128 while that of the discriminator is 131:1024:2048:1.

3.2.2 Hyper-parameter Settings

The learning rate for the generator is set to 2×10^{-4} , while for the discriminator is set to 1×10^{-4} . These were selected empirically after some preliminary tests. The Adam optimizer was used with key parameters $\beta_1 = 0.9$, $\beta_2 = 0.9999$ in both the discriminator and the generator. A maximum number of $K = 20,000$ iterations were considered. The V-matrix regularization coefficient γ is discussed in Sect. 4.1.4.

3.2.3 Considered Scenarios

Several scenarios were taken into consideration to assess the effectiveness of the proposed model. In all these scenarios, all the hyper-parameter of each GAN are the same as mentioned before except for the loss function. Besides, we use the additive form (29) of V-matrix in VGAN. We used an SVM (support vector machine) classifier with an RBF (radial basis function) kernel for fault classification. For early stopping, we use a random forest for selecting G_{\max} (39). The different scenarios are:

- (i) *normal*, i.e., cWGAN-GP loss (19) and early stopping, i.e., the employed generator given by (39);
- (ii) *normal_l*, i.e., cWGAN-GP without early stopping, i.e., the employed generator corresponding to the generator obtained in the last iteration;
- (iii) *mse*, i.e., MSE GAN (33) and early stopping;
- (iv) *mse_l*, i.e., MSE GAN without early stopping;
- (v) *v*, i.e., VGAN (32) and early stopping;
- (vi) *v_l*, i.e., VGAN without early stopping.

3.3 Experimental Settings of SW-CycleGAN

In the experiment of SW-CycleGAN, six different fault conditions were taken into consideration. These conditions are (1) Health condition (C_0); (2) Cracking in Planetary gear A (C_4); (3) Broken tooth in Planetary gear A (C_5); (4) Broken tooth in Sun gear A (C_6); (5) Cracking in Planetary gear B (C_7) and (6) Broken tooth in Planetary gear B (C_8) (See Table 1).

3.3.1 Neural Network Architectures

The neural network architectures of the generator and the discriminator in both CycleGAN and conditional CycleGAN are shown in Fig. 9. Unlike the image dataset that uses 2-dimensional CNN for computation, the robot data set resorts to 1-dimensional CNN (1D-CNN). For instance, if the input vector has N features, it needs to be reshaped to the shape of (N,1) to be suitable for the 1D-CNN. Therefore, the input for the generator and the discriminator is a series of reshaped vectors.

As shown in Fig. 9, in the unconditional generator, the input images went through two CNN blocks (CNN block down 1), two modified ResNet blocks for repeating 3 times to enhance features and then these features are sent to CNN blocks (CNN block up) with two modified ResNet blocks. Repeat this operation two times to get the output of these blocks. Finally, these output went though one CNN block up and one Transpose CNN for generating synthetic images. The activation of the last layer in the generator is the tanh. For the architecture of the unconditional discriminator, the input is the same as the generator as it also receives digit images. The input goes through seven CNN blocks (CNN block down 2) with both instance normalization and leaky Relu. The output is flattened through one dense layer to get logits. Moreover, the details of CNN block down 1, CNN block down 2 and CNN block up are given in Fig. 9c Block descriptions.

The architectures for the conditional CycleGAN are similar to the unconditional one just described except for the inputs that were extended with labels. The features are extended with labels to sent to three CNN blocks (CNN block down 1) and went through four Modified ResNet blocks. Then the output is sent to two CNN blocks (CNN block up) and went through one Transpose CNN with tanh activation to generate synthetic data (see Fig. 9a Generator). For the discriminator, the input will go through six CNN blocks (CNN block down 2) and then flattened to get logits (Fig. 9a Discriminator). Leaky relu activations are used in every convolutional layer except for the last layer of the generators. In contrast, the relu activation was used for the convolutional layer in the discriminators as no vanished gradient was observed for such modules.

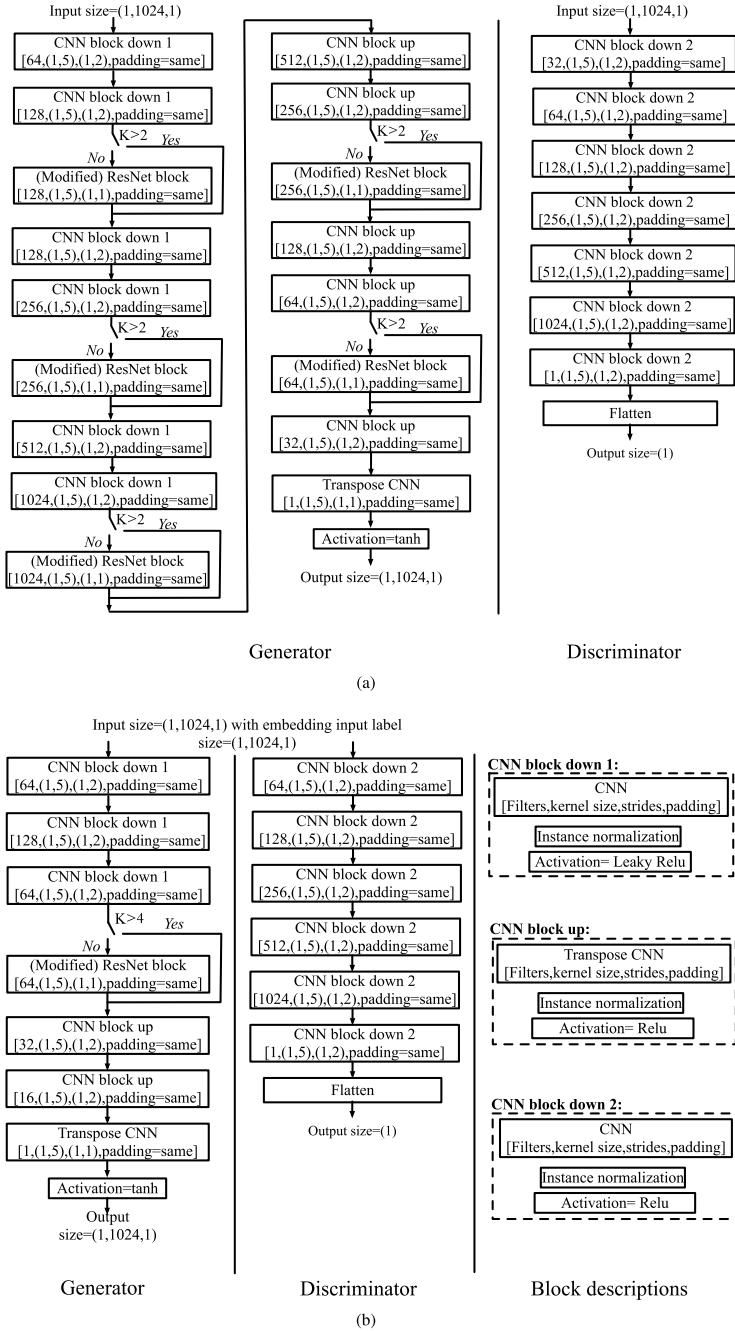


Fig. 9 The architecture of CycleGAN for the industrial robot data set: **a** unconditional CycleGAN; **b** conditional CycleGAN

3.3.2 Hyper-parameter Settings

In the experiment of CycleGAN and conditional CycleGAN, a sliced Wasserstein distance with 32 random projections ($r = 32$) was considered in the generator loss function. The L_2 norm is also considered in the cycle consistency loss with the λ_c set to 10. The discriminator and generator learning rates were $1e-4$ and $2e-5$, respectively. All modules were being optimized with Adam. The maximum iterations of unconditional and conditional CycleGAN were set to 10,000 and 18,000, respectively.

3.3.3 Considered Scenarios

Four scenarios were taken into consideration:

- (i) wd, i.e., Wasserstein distance loss in both CycleGAN and conditional CycleGAN with the neural network architecture of the ResNet;
- (ii) wd-sem, i.e., Wasserstein distance loss in both CycleGAN and conditional CycleGAN with the neural network architecture of the modified ResNet;
- (iii) swd, i.e., sliced Wasserstein distance loss in both CycleGAN and conditional CycleGAN with the neural network architecture of the ResNet;
- (iv) swd-sem, i.e., sliced Wasserstein distance loss in both CycleGAN and conditional CycleGAN with the neural network architecture of the modified ResNet.

4 Results and Discussion

4.1 Results on VGAN

In this subsection, results on the above mentioned issues such as the imbalanced data set problem, the convergence problem, and the sensitivity of the algorithm to relevant hyper-parameters are presented. The subsequent discussion takes into account confusion matrices and statistical tests of hypotheses.

4.1.1 Comparisons of the Different Scenarios

Figure 10 shows the distribution of the obtained accuracy for each scenario using boxplots. A boxplot summarizes a data distribution stressing five characteristic values: minimum, lower quartile, median, upper quartile, and maximum value. The red line denotes the median value. The distribution pertains to 20 independent repetitions. The results presented in Fig. 10 were analyzed by the Friedman test, a non-parametric statistic test of hypotheses, to evaluate whether or not there is a statistically significant difference between the results of the different scenarios. The Friedman null

Fig. 10 Boxplots exhibiting the distribution of accuracy over 30 independent runs for the different scenarios

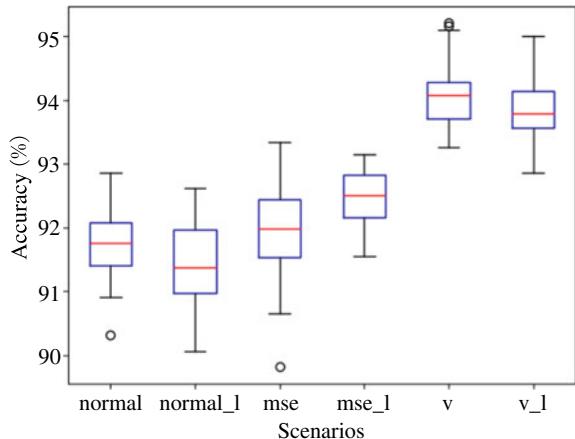


Table 2 The number of examples available for each operating condition. C_0 stands for the class of nominal operating state, while C_1 , C_2 , and C_3 represents faulty states classes

Class	C_0	C_1	C_2	C_3
Training set	14000	140	140	140
Validation set	6000	6000	6000	6000

hypothesis is that there is no statistically significant difference between the results of the different scenarios. Given a significant level, α , this hypothesis cannot be rejected whenever the p_{Friedman} , the p -value generated by the test, satisfies $p_{\text{Friedman}} > \alpha$. The null hypothesis is rejected otherwise, meaning that there is a statistically significant difference between the analyzed scenarios. In such a case, we can detect which of the scenario is responsible for such a difference by resorting to a pairwise *posthoc* test. A ranking can be obtained by counting the number of times that a method was a winner in the pairwise comparison. See [53, 54] for further details. Here we use the usual $\alpha = 0.05$ and the Wilcoxon test as *posthoc*.

As the Friedman test for the results presented in Fig. 10 yielded $p_{\text{Friedman}} = 8.79 \times 10^{-61} < 0.05$, we must conclude that a statistically significant difference exists between the six scenarios. Consequently, we present in Table 5 the obtained Wilcoxon *post hoc* results.

From Table 3 one can see that scenario v (VGAN with early stopping) has 5 wins meaning that it ranks first among all the others.

From Fig. 10 we observe that median accuracy is (i) normal: 91.75%; (ii) normal_l: 91.483%; (iii) mse: with 91.961%; (iv) mse_l: 92.47%; (v) v: 92.93%; and (vi) v_l: 92.36%. Also, with or without early stopping, VGAN outperforms all the other scenarios. More specifically, scenario (v) has an accuracy 1.169% higher than the normal scenario (i).

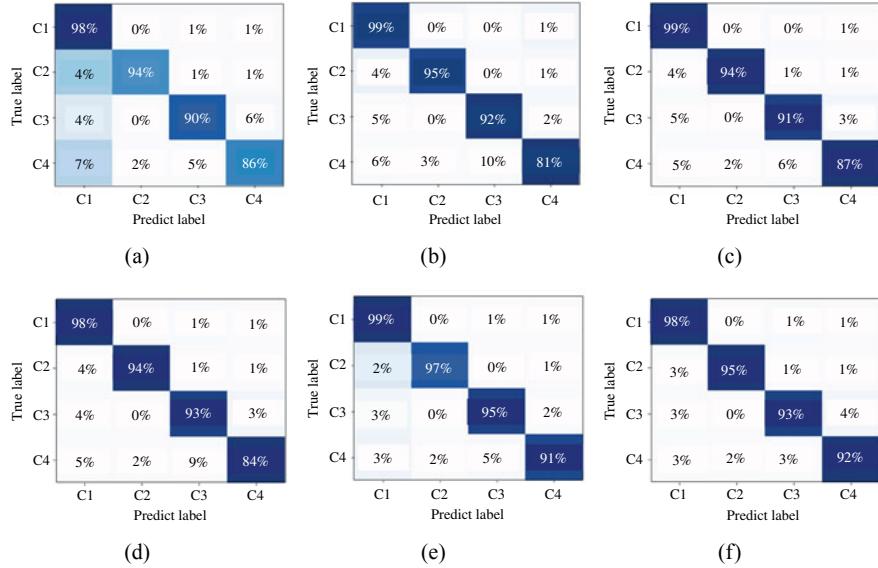


Fig. 11 Confusion matrices for scenario **a** Normal; **b** Normal_l; **c** mse; **d** mse_l; **e** v; and **f** v_l

The confusion matrices for the different scenarios are presented in Fig. 11. As observed in Fig. 11, all these matrices concern the 6000 validation set mentioned in Table 2. In each matrix the diagonal refers to the correct classification rate in the corresponding class. All the other entries refer to mis-classifications. From these figures one can see also that confusion matrix for scenario v (VGAN with early stopping) has the best classification performance.

To further illustrate the performance, t-Distributed Stochastic Neighbour Embedding (t-SNE) is used to characterize the fault data generation that is represented in Fig. 12. As shown in Fig. 12, the green part is C_1 , the blue part is C_2 and the rest part is C_3 . From Fig. 12, Both of the clustering performance is worse except Fig. 12e, which is our proposal in this work. This is an evidence that our data generation method is more appropriate for fault data generation than any other model.

Besides, the calculation burden of the above scenarios is presented in Table 4, it can be found that with our early stopping and the V-matrix based loss function, this model can quickly find the desired generator for optimal data generation.

4.1.2 Convergence Curves

Figure 13 shows typical convergence curves for scenario v (VGAN with early stopping). As shown in Fig. 13 the loss curves of both discriminator and generator are still oscillating. We hypothesize that this is an inevitable side effect of adversarial learning between the generator and the discriminator. Notice however that, without

Table 3 Wilcoxon post-hoc results for the six studied scenarios

Comparison	p-value	Winner
Normal versus normali_l	0.032	Normal
Normal versus mse	0.0198	—
Normal versus mse_l	6.88×10^{-5}	mse_l
Normal versus v	1.7×10^{-6}	v
Normal versus v_l	1.73×10^{-6}	v_l
Normal versus mse	0.02	mse
Normal versus mse_l	9.77×10^{-6}	mse_l
Normal versus v	1.73×10^{-6}	v
Normal versus v_l	1.73×10^{-6}	v_l
mse versus mse_l	0.01	mse_l
mse versus v	1.73×10^{-6}	v
mse versus v_l	1.73×10^{-6}	v_l
mse_l versus v	1.73×10^{-6}	v
mse_l versus v_l	$1.7 \times 10^{-6}3$	v_l
v versus v_l	0.0125	v

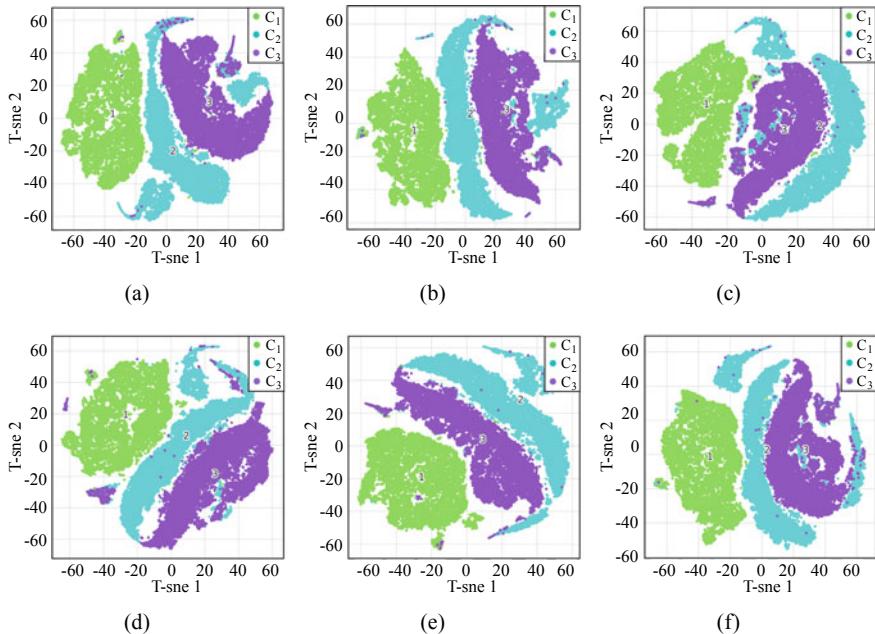
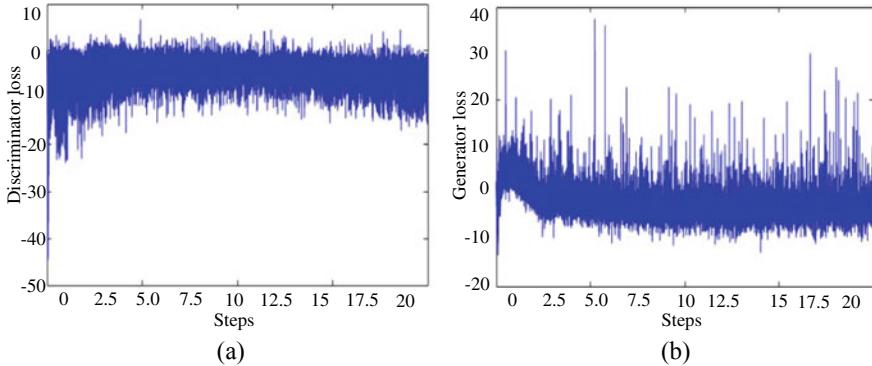
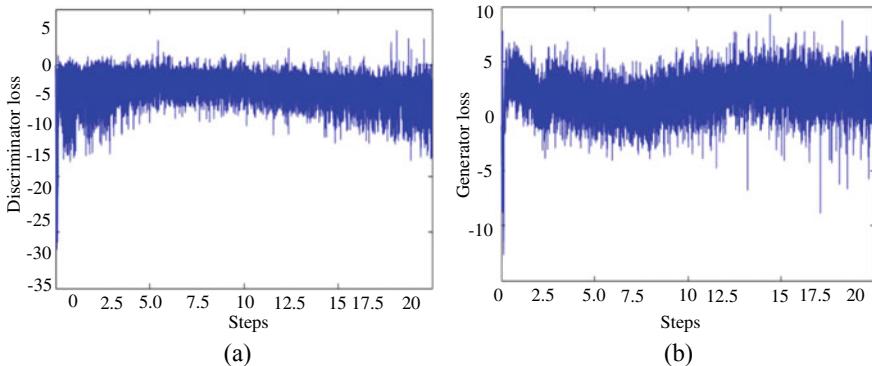
**Fig. 12** t-SNE representation of each data generation scenario: **a** normal; **b** normal_l; **c** mse; **d** mse_l; **e** v and **f** v_l

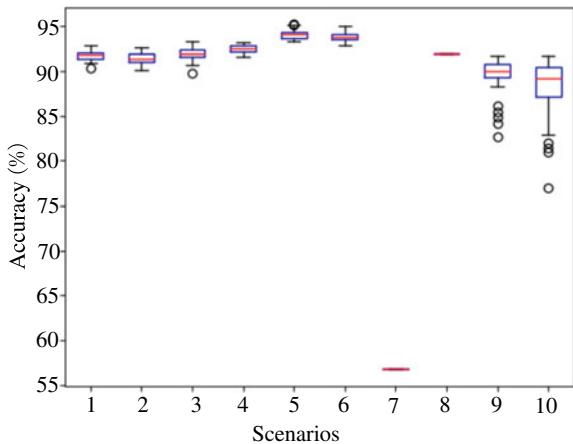
Table 4 Calculation burden via different scenario for fault data generation

Scenario	Normal	Norma_1	mse	mse_1	v	v_1
Times (s)	2905	3152	2463	3092	2138	3653

**Fig. 13** Typical convergence curves for VGAN with early stopping: **a** discriminator and **b** generator**Fig. 14** Typical convergence curves for cWGAN-GP without early stopping: **a** discriminator and **b** generator

V-regularization, oscillations are way worse: Fig. 14 shows for the non-regularized case that while the loss evolution looks similar to Fig. 13 the loss ranges are much smaller ($[-35; 5]$ versus $[-50; 10]$ for the discriminator; $[-15; 10]$ versus $[-20; 40]$ for the generator).

Fig. 15 Results of the difference scenarios for dealing with the imbalance training data; see text for details



4.1.3 On Imbalance Data Sets

To assess the effectiveness of the model when dealing with an imbalanced data set, ten scenarios were studied, as follows:

- 1: normal;
- 2: normal_l;
- 3: mse;
- 4: mse_l;
- 5: v;
- 6: v_l (all the above with the previous presented meaning);
- 7: imbalance training set, i.e., 14,000 health examples and only 1% of examples (140) for each fault;
- 8: imbalance training set, as above, but processed by the SMOTE technique;
- 9: vanilla unsupervised GAN with early stopping as per [55];
- 10: vanilla unsupervised GAN without early stopping;

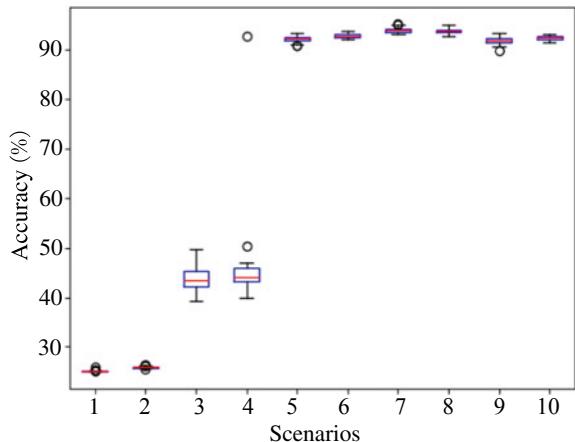
In all these scenarios, an SVM classifier with the RBF kernel was used for fault classification.

From Fig. 15, we can see that the 5th model (VGAN with early stopping) has the highest median accuracy. The Friedman test reveals a statistical significant difference $p_{\text{Friedman}} = 0.012 < \alpha = 0.05$.

4.1.4 On the Model Sensitivity

Below we present a sensitivity discussion on four relevant aspects of the model: regularization term, ratios between healthy and faulty data, weight initialization, and accuracy versus epochs.

Fig. 16 Accuracy values for the v model (VGAN with early stopping) and v_1 model (VGAN without early stopping) for different values of γ in (32)



Regularization term: To evaluate the effect of the V-regularization coefficient γ in (32), we compared a v model (VGAN with early stopping) with v_1 (VGAN without early stopping). Both scenarios share the same hyper-parameter settings except for γ that takes values in $\{1 e^{-1}, 1 e^{-2}, 1 e^{-3}, 1 e^{-4}, 0\}$ see Fig. 16. In this figure, 1 to 10 stands for ve1, ve1_1, (), ve2, ve2_1, ve3, ve3_1, ve4, ve4_1, 0, 0_1, respectively. Besides, Labels ve1, ve1_1 correspond to $\gamma = 1 e^{-1}$, ..., ve4, ve4_1 to $\gamma = 1 e^{-4}$, while labels 0 and 0_1 correspond to $\gamma = 0$ in a v and v_1 model, respectively.

The value of $\gamma = 1 e^{-4}$ yielded the highest accuracy values among all other cases, including the case where $\gamma = 0$ corresponds to the cWGAN-GP model (19). Another observation is that early stopping is a better option.

Friedman and Wilcoxon *post hoc* tests are also performed for evaluating the effect of the regularization term. The ten scenarios in Fig. 16 yield $p_{\text{Friedman}} = 6.21 \times 10^{-61}$ meaning that there exists a significant difference among them. Afterward, the *post hoc* analysis using Wilcoxon test shows that the regularization term $\gamma = 1 e^{-4}$ with early stopping (ve4 model) with 9 wins ranks first among all the others.

Faulty versus healthy data ratios: To illustrate the importance of using a balanced data set in this application, we evaluate the performance of the SVM fault classifier under different faulty to healthy data ratios. As always, the number of healthy examples in the training set is 14,000 and the initial number of examples in each fault class is 140 as per Table 2. The following percentages were considered $perc = 1, 2, 4, 6, 8, 10, 20, 40, 60, 80$, and 100%. For instance, when $perc=10\%$, only $14,000 \times 10\% = 1400$ examples were used for each fault class. From these $1400 - 140 = 1260$ examples were generated with our best model (VGAN with early stopping). The resulting learning curve is shown in Fig. 17. The average accuracy increased from 56.25 to 95.17% by increasing the availability of faulty data from 1% (140 examples) to 100% (14,000 examples). That is, the imbalanced data set had a strong negative impact on the fault classifier. Furthermore, there was a 20%+ increase in performance in the range $1\% \leq perc \leq 40\%$; afterward the improvement in accu-

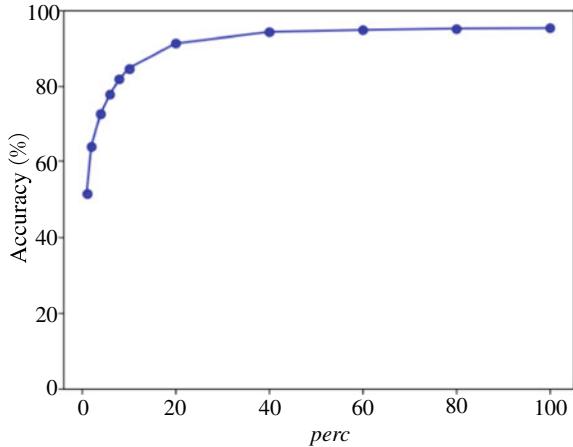


Fig. 17 A learning curve taking into account a percentage $perc$ of the faulty examples relatively to the number of healthy examples

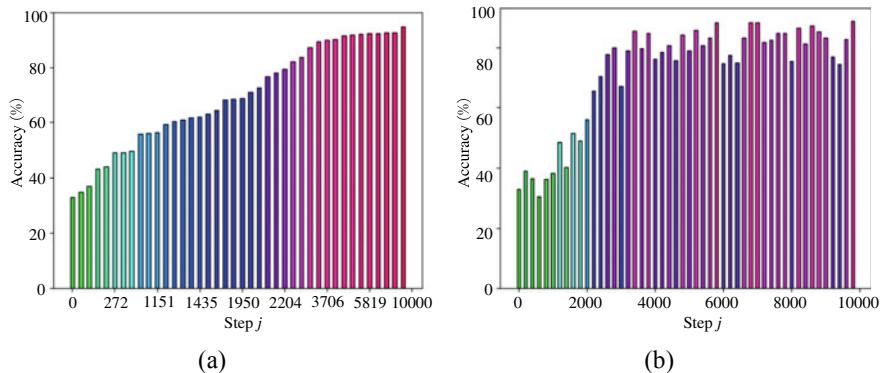


Fig. 18 Classification accuracy obtained for VGAN using (39) as generator, as a function of the epochs

racy was slower and slower. For $perc \geq 80\%$ the improvement was neglectable. That is, adding more data after that point hardly improved the performance.

Weight initialization: In our experiments we considered 30 different random seeds for weight initialization. The impact of the different initial weights are reflected in the dispersion of accuracy as presented in the boxplots of the above figures.

Accuracy versus epochs: Figure 18a shows the classification accuracy obtained using (39) as a generator as a function of the current number of epochs for VGAN. From Fig. 18 as the number of epochs increased a monotonic increase in the accuracy is observed, from 38% up to 93.28%.

When the model v_1 (VGAN without early stopping) is used in Fig. 18b, we can see that the accuracy in each epoch is not monotonic increasing.

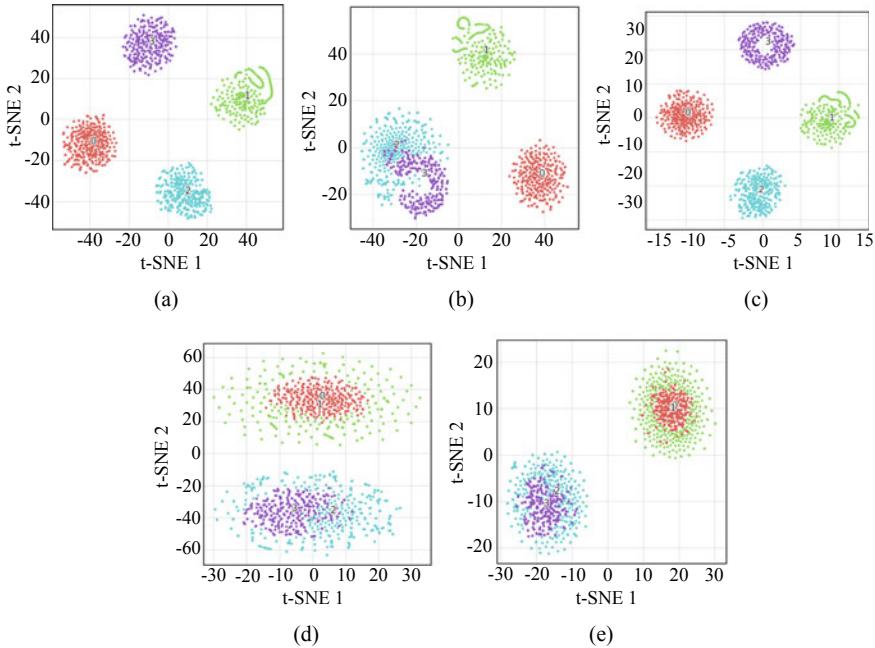


Fig. 19 t-SNE visualization with the industrial robot dataset: **a** initial state; **b** wd; **c** wd + sem; **d** swd and **e** swd + sem

4.2 Results on SW-CycleGAN

4.2.1 Unconditional CycleGAN

Figure 19 shows a two-dimensional visualization of the (high-dimensional) feature space using t-distributed stochastic neighbor embedding (t-SNE) [56]. In this figure, the number 1 (in red) stands for the nominal condition C_0 . In contrast, the number 2 (in green) stands for condition C_1 . Moreover, numbers 0 and 3 denote the generated data of C_0 and C_1 , respectively. The objective is to let 0 and 3 close to 1 and 2 as much as possible. That would correspond to a successful data transference from the source to the target domain. From this figure, it is apparent that the SWD-based CycleGAN scenarios (swd and swd-sem) are performing better when compared to the vanilla CycleGAN (scenarios wd and wd-sem). The modified residual network scenario (swd-sem) appears to outperform the others.

For the sake of statistical analysis, 30 independent runs (with different random weights) were executed for each scenario. The MC rate is used to measure the performance of each scenario. The results obtained for each scenario are presented in the boxplots in Fig. 20. A boxplot consists of the min, the median (red line), and the max values, as well as outliers, of a given data sample. From this figure, one can

Fig. 20 The boxplot of different scenarios with the industrial robot dataset

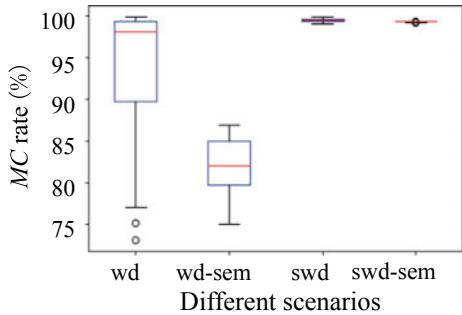
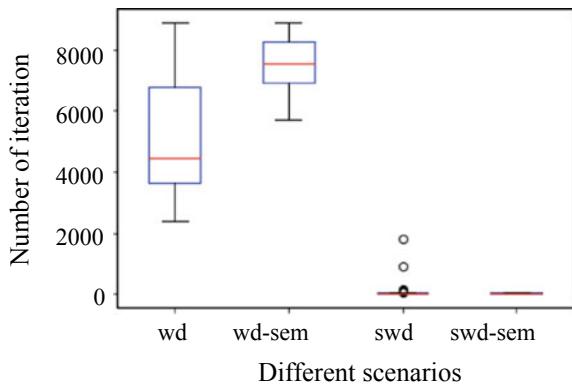


Fig. 21 Dispersion of the required number of iterations to reach the same level of performance, over 30 independent runs in the industrial robot data set



be found that the industrial robot dataset, the median MC rate is (i) wd: 93.22%; (ii) wd-sem: 81.90%; (iii) swd: 99.40%; and (vi) swd-sem: 99.21%.

For accessing training efficiency, a threshold on the MC rate was set to check the minimum iterations required to reach such a threshold. MC thresholds of 99.50% for the industrial robot data set were set. The results shown in Fig. 21 area. For the industrial robot data set, we observed that to reach $MC = 99.5\%$, the average number of iterations in each scenario are (i) wd: 7561; (ii) wd-sem: 5078; (iii) swd: 130; and (vi) swd-sem: 39.

This is, SWD-based models need way fewer iterations to achieve the same performance level when compared with their counterparts equipped with WD. In the case of the robot, scenario swd required 1 (one) order of magnitude less iterations than the original one (scenario wd) while scenario swd-sem required 2 (two) orders of magnitude fewer iterations than the corresponding one (wd-sem).

To quantitatively evaluate the results of Fig. 20, a statistical analysis was performed. The Friedman test is used for checking possible differences among the distributions. If such a difference exists, then a *post hoc* test, i.e., the Wilcoxon test, is used for ranking (see [54] for details). Friedman test yielded a $p_{Friedman} = 2.4 \times 10^{-13}$ in the industrial robot data set, revealing that the null-hypothesis must be rejected

Table 5 Wilcoxon *post-hoc* results of unconditional CycleGAN for the four studied scenarios

Dataset	Comparison	<i>p</i> -value	Winner
Industrial robot	wd versus wd-sem	8.167×10^{-5}	wd
	wd versus swd	5.25×10^{-6}	swd
	wd versus swd-sem	6.914×10^{-4}	swd-sem
	wd-sem versus swd	3.78×10^{-6}	swd
	wd-sem versus swd-sem	3.78×10^{-6}	swd-sem
	swd versus swd-sem	1.88×10^{-4}	swd-sem

(i.e., there is a statistically significant difference among samples) and therefore *post hoc*s were ran whose results are shown in Table 5.

From Table 5 shows that scenario swd-sem has three wins for both the MNSIT and the industrial robot data sets, ranking first among other scenarios. Do this relative performance also hold for the conditional cycleGAN?

4.2.2 Conditional CycleGAN

Figure 22 present t-SNE projection of real feature space into the plan. Figure 22a shows the generated data for G1 to G5 are distinct from the real data at the beginning of the training. Figure 22b–e show the obtained results after training for different scenarios. From the result presented in the figure, it is apparent that swd-sem outperforms all the others.

Figure 23 shows the dispersion of the *MC* rate obtained from 30 independent runs. From this, we can find that for the the industrial robot data set, the median *MC* rate is (i) wd: 98.96%; (ii) wd-sem: 99.29%; (iii) swd: 99.69%; and (vi) swd-sem: 99.73%.

Since SWD has a faster convergence rate than WD in unconditional CycleGAN, considering its generalization performance, we further evaluate this phenomenon in conditional CycleGAN as shown in Fig. 24. A threshold of *MC*=99.5% for the robot was set. As shown in Fig. 24, the average number of iterations required were (i) wd: 16,249; (ii) wd-sem: 15,637; (iii) swd: 13,668; and (vi) swd-sem: 13,034.

Again, SWD-based models needed fewer iterations than their traditional counterparts; not as notorious as the unconditional case, though.

Statistical tests of hypotheses yielded the results displayed in Table 6. From the above analysis, one can find that swd-sem outperforms all the others for both MNIST and the industrial robot data sets.

Besides, the convergence curve of both generator and discriminator on the industrial robot data sets is shown in Fig. 25. Here, scenarios swd-sem and wd-sem are introduced to see the phenomenon. It is clearly shown that scenario swd-sem has a smother curve than wd-sem.

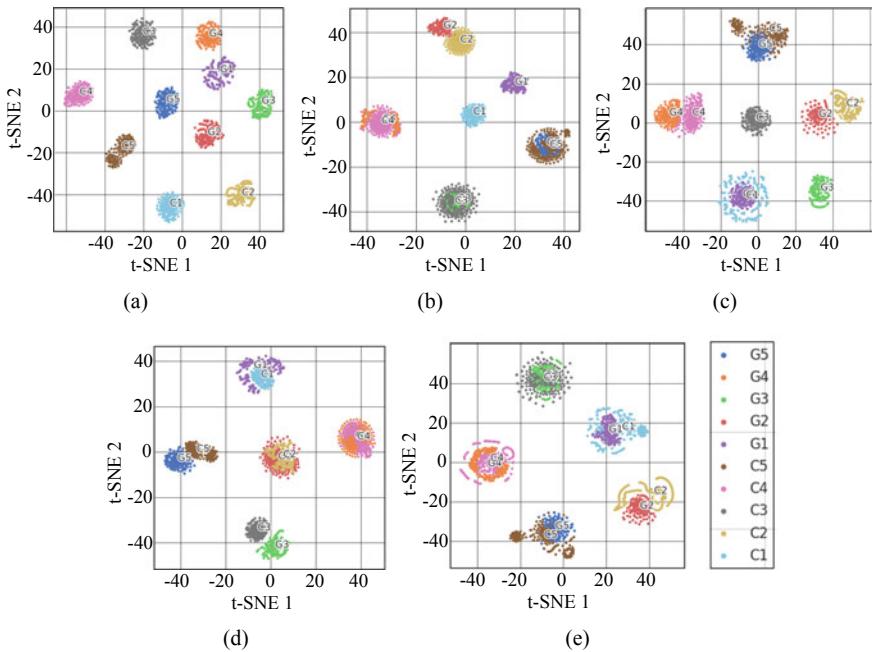


Fig. 22 t-SNE visualization with the industrial robot data set: **a** initial learning state; **b** wd; **c** wd-sem; **d** swd and **e** swd-sem

Fig. 23 Boxplots for different scenarios in the industrial robot

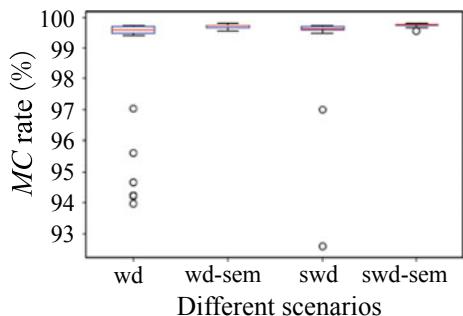
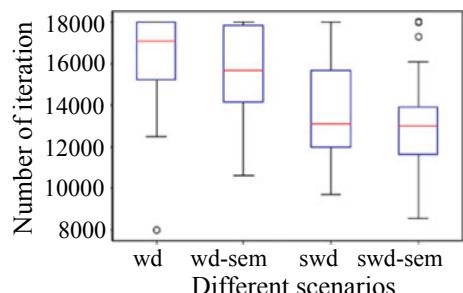


Fig. 24 The convergence curves for the different scenarios in the industrial robot data set



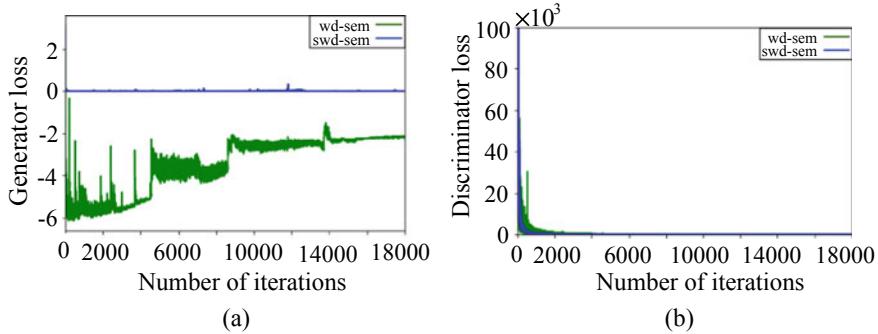


Fig. 25 Typical convergence curve on the industrial robot data set for **a** Generator; **b** Discriminator

5 Conclusion

Generative adversarial networks (GANs) can be used for data augmentation in data-driven fault diagnosis, where data from the nominal state are relatively abundant while data from the different faulty states can be expensive to acquire, if possible. Without a balanced data set for training, fault classifiers simply do not have an acceptable out-of-sample performance.

One of the main problems in practical applications of GANs is the oscillatory behavior of loss during training and mode collapse. The loss function used plays a crucial role in the first issue. Mode collapse can be addressed by resorting to CycleGANs, a combination of two GANs used in such a way that for each input, a different synthetic example should be generated and from which the original input can be reconstructed.

With this in mind, we presented VGAN, a V-regularized conditional Wasserstein GAN with gradient penalization. V-regularization is based on a criterion that generalizes the mean square error criterion in the sense that it takes into account not only residuals but also the dispersion among the independent variable in input data.

Table 6 Wilcoxon post-hoc results for the four studied scenarios with conditional CycleGAN

Dataset	Comparison	p-value	Winner
Industrial robot	wd versus wd-sem	1.4×10^{-2}	wd
	wd versus swd	1.45×10^{-5}	swd
	wd versus swd-sem	2.84×10^{-6}	swd-sem
	wd-sem versus swd	2.9×10^{-4}	swd
	wd-sem versus swd-sem	2.847×10^{-6}	swd-sem
	swd versus swd-sem	1.7×10^{-4}	swd-sem

VGAN generalizes both MSE GAN and cWGAN-GP. A novel early stopping like strategy was also proposed.

Also, we have shown that to transfer abundant normal state data to the different scarce faulty data, the employment of sliced Wasserstein distance in CycleGANs (SW-CycleGAN) resulted in a smoother, faster, and more efficient training process when compared with the traditional Wasserstein CycleGAN.

The effectiveness of the presented formalisms was illustrated through a comprehensive set of experiments on a fault diagnosis task for an industrial robot. Results on VGAN show that the VGAN outperforms nine other scenarios including vanilla GAN, conditional WGAN-GP with and without MSE based regularization, and SMOTE, a classic data augmentation technique. Furthermore, the early stopping like mechanism allows one to obtain a monotonic increasing performance of the model during training and, when combined with the V-regularization, ranks first in terms of classification accuracy among all other scenarios. Results on SW-CycleGAN show that for both the unconditional and the conditional cases, sliced Wasserstein distance outperforms classic Wasserstein distance in CycleGANs. A model compatibility of 99.73% (conditional case) and 99.21% (unconditional case) were observed. In some cases, the improvement in convergence efficiency was higher than 2 (two) orders of magnitude.

Acknowledgements This work is supported in part by Portuguese funds through FCT-Foundation for Science and Technology, in part by I.P., through IDMEC, under LAETA Project UIDB/50022/2020, in part by the National Natural Science Foundation of China under Grant 52175080, in part by the Chongqing Natural Science Foundation under Grant cstc2019jcyj-zdxmX0013, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2019B1515120095, and in part by the Intelligent Manufacturing PHM Innovation Team Program under Grant 2018KCXTD029.

References

1. Iqbal, J., Islam, R. U., Abbas, S. Z., Khan, A. A., & Ajwad, S. A. (2016). Automating industrial tasks through mechatronic systems-a review of robotics in industrial perspective. *Tehnički vjesnik*, 23(3), 917–924.
2. Caccavale, F., Cilibrizzi, P., Pierri, F., & Villani, L. (2009). Actuators fault diagnosis for robot manipulators with uncertain model. *Control Engineering Practice*, 17(1), 146–157.
3. Ali, J. B., Fnaiech, N., Saidi, L., Chebel-Morello, B., & Fnaiech, F. (2015). Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals. *Applied Acoustics*, 89, 16–27.
4. Yan, K., Ji, Z., Lu, H., Huang, J., Shen, W., & Xue, Y. (2017). Fast and accurate classification of time series data using extended elm: Application in fault diagnosis of air handling units. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(7), 1349–1356.
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139–144.
6. Zhang, S. (2022). *On the nash equilibrium of moment-matching gans for stationary Gaussian processes*. ArXiv, abs/2203.07136.

7. Premkumar, M., Jangir, P., Ramakrishnan, C., Nalinipriya, G., Alhelou, H. H., & Kumar, B. S. (2021). Identification of solar photovoltaic model parameters using an improved gradient-based optimization algorithm with chaotic drifts. *IEEE Access*, 9, 62347–62379.
8. Koziel, S., & Pietrenko-Dabrowska, A. (2021). Accelerated gradient-based optimization of antenna structures using multifidelity simulations and convergence-based model management scheme. *IEEE Transactions on Antennas and Propagation*, 69(12), 8778–8789.
9. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. In: *Advances in neural information processing systems* (Vol. 29).
10. Lee, P. Y., Hui, S. C., & Fong, A. C. M. (2002). Neural networks for web content filtering. *IEEE Intelligent Systems*, 17(5), 48–57.
11. Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1), 79–86.
12. Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. In *International Conference on Machine Learning* (pp. 214–223). PMLR.
13. Levine, A., & Feizi, S. (2020). Wasserstein smoothing: Certified robustness against wasserstein adversarial attacks. In *International Conference on Artificial Intelligence and Statistics* (pp. 3938–3947). PMLR.
14. Donahue, J., & Krähenbühl, P., & Darrell, T. (2016). *Adversarial feature learning*. arXiv preprint [arXiv:1605.09782](https://arxiv.org/abs/1605.09782).
15. Yang, S., Xie, L., Chen, X., Lou, X., Zhu, X., Huang, D., & Li, H. (2017). Statistical parametric speech synthesis using generative adversarial networks under a multi-task learning framework. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (pp. 685–691). IEEE.
16. Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *Computing Research Repository*, page [arXiv:1411.1784](https://arxiv.org/abs/1411.1784).
17. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of wasserstein gans. In *Advances in neural information processing systems* (Vol. 30).
18. Li, W., Zhong, X., Shao, H., Cai, B., & Yang, X. (2022). Multi-mode data augmentation and fault diagnosis of rotating machinery using modified acgan designed with new framework. *Advanced Engineering Informatics*, 52, 101552.
19. Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., & Paul Smolley, S. (2017). Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2794–2802).
20. Lim, J. H., & Ye, J. C. (2017). *Geometric gan*. ArXiv, abs/1705.02894.
21. Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). *Spectral normalization for generative adversarial networks*. CoRR, abs/1802.05957.
22. Tran, D., Ranganath, R., & Blei, D. (2017). Hierarchical implicit models and likelihood-free variational inference. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc.
23. Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2223–2232).
24. Che, T., Li, Y., Jacob, A. P., Bengio, Y., & Li, W. (2016). *Mode regularized generative adversarial*.
25. Vapnik, V., & Izmailov, R. (2015). V-matrix method of solving statistical inference problems. *Journal of Machine Learning Research*, 16(51), 1683–1730.
26. Vapnik, V., & Izmailov, R. (2019). Rethinking statistical learning theory: learning using statistical invariants. *Machine Learning*, 108(3), 381–423.
27. Pu, Z., Cabrera, D., Li, C., & de Oliveira, J. V. (2022). VGAN: Generalizing MSE GAN and WGAN-GP for robot fault diagnosis. *IEEE Intelligent Systems*, 37(3), 65–75.
28. Nadjahi, K. (2021). *Sliced-Wasserstein distance for large-scale machine learning: theory, methodology and extensions*. Ph.D. thesis, Institut polytechnique de Paris.

29. Deshpande, I., Zhang, Z., & Schwing, A. (2018). Generative modeling using the sliced Wasserstein distance. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3483–3491). Los Alamitos, CA, USA: IEEE Computer Society.
30. Zhao, K., Jiang, H., Liu, C., Wang, Y., & Zhu, K. (2022). A new data generation approach with modified wasserstein auto-encoder for rotating machinery fault diagnosis with limited fault data. *Knowledge-Based Systems*, 238, 107892.
31. Pu, Z., Cabrera, D., Li, C., & de Oliveira, J. V. (2023). Sliced wasserstein cycle consistency generative adversarial networks for fault data augmentation of an industrial robot. *Expert Systems with Applications*, 222, 119754.
32. Wang, J., Han, B., Bao, H., Wang, M., Chu, Z., & Shen, Y. (2020). Data augment method for machine fault diagnosis using conditional generative adversarial networks. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 234(12), 2719–2727.
33. Balaji, Y., Min, M. R., Bai, B., Chellappa, R., & Graf, H. P. (2019). Conditional gan with discriminative filter generation for text-to-video synthesis. *IJCAI*, 1, 2.
34. Hu, T., Long, C., & Xiao, C. (2021). A novel visual representation on text using diverse conditional gan for visual recognition. *IEEE Transactions on Image Processing*, 30, 3499–3512.
35. Dash, A., Gamboa, J. C. B., Ahmed, S., Liwicki, M., & Afzal, M. Z. (2017). Tac-gan - text conditioned auxiliary classifier generative adversarial network. *CoRR*, abs/1703.06412.
36. Gauthier, J. (2014). Conditional generative adversarial nets for convolutional face generation. In *Class project for Stanford CS231N: Convolutional neural networks for visual recognition, Winter semester* (Vol. 2014, no. 5, p. 2).
37. Gu, S., Bao, J., Yang, H., Chen, D., Wen, F., & Yuan, L. (2019). Mask-guided portrait editing with conditional gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3436–3445).
38. Tang, H., Xu, D., Sebe, N., Wang, Y., Corso, J. J., & Yan, Y. (2019). Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2417–2426).
39. Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. Towards diverse and natural image descriptions via a conditional gan. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2970–2979).
40. Wu, J., Wang, Y., Xue, T., Sun, X., Freeman, B., & Tenenbaum, J. (2018). 3d-aware scene manipulation via inverse graphics. In *Advances in neural information processing systems* (Vol. 31).
41. Yang, Z., Chen, W., Wang, F., & Xu, B. (2017). *Improving neural machine translation with conditional sequence generative adversarial nets*. arXiv preprint [arXiv:1703.04887](https://arxiv.org/abs/1703.04887).
42. Shapeev, V., Golushko, S., Belyaev, V., Bryndin, L., & Kirillov, P. (2021). New versions of the least-squares collocation method for solving differential and integral equations. *Journal of Physics: Conference Series*, 1715, 012031. IOP Publishing.
43. Kodali, N., Abernethy, J., Hays, J., & Kira, Z. (2018). On convergence and stability of gans. In *International Conference on Learning Representations*.
44. Freirich, D., Michaeli, T., & Meir, R. (2021). A theory of the distortion-perception tradeoff in wasserstein space. *Advances in Neural Information Processing Systems*, 34.
45. Piradl, S., & Shadrokh, A. (2021). Robust minimum distance estimation of a linear regression model with correlated errors in the presence of outliers. *Communications in Statistics-Theory and Methods*, 50(23), 5488–5498.
46. Deshpande, I., Zhang, Z., & Schwing, A. G. (2018). Generative modeling using the sliced wasserstein distance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3483–3491).
47. Lee, C. Y., Batra, T., Baig, M. H., & Ulbricht, D. (2019). Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10285–10295).

48. Nadjahi, K., Durmus, A., Jacob, P. E., Badeau, R., & Simsekli, U. (2021). Fast approximation of the sliced-wasserstein distance using concentration of random projections. *Advances in Neural Information Processing Systems*, 34.
49. Tang, H., Liu, H., Xu, D., Torr, P. H., & Sebe, N. (2021). Attentiongan: Unpaired image-to-image translation using attention-guided generative adversarial networks. *IEEE Transactions on Neural Networks and Learning Systems*.
50. Lu, Y., Tai, Y. W., & Tang, C. K. (2018). Attribute-guided face generation using conditional cyclegan. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 282–297).
51. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).
52. Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7132–7141).
53. Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7, 1–30.
54. Pacheco, F., de Oliveira, J. V., Sánchez, R. V., Cerrada, M., Cabrera, D., Li, C., Zurita, G., & Artés, M. (2016). A statistical comparison of neuroclassifiers and feature selection methods for gearbox fault diagnosis under realistic conditions. *Neurocomputing*, 194, 192–206.
55. Cabrera, D., Sancho, F., Long, J., Sánchez, R. V., Zhang, S., Cerrada, M., & Li, C. (2019). Generative adversarial networks selection approach for extremely imbalanced fault diagnosis of reciprocating machinery. *IEEE Access*, 7, 70643–70653.
56. Han, Y., Liu, S., Cong, D., Geng, Z., Fan, J., Gao, J., & Pan, T. (2021). Resource optimization model using novel extreme learning machine with t-distributed stochastic neighbor embedding: Application to complex industrial processes. *Energy*, 225, 120255.

Underwater Acoustic Noise Modeling Based on Generative-Adversarial-Network



Junfeng Wang, Mingzhang Zhou, Yue Cui, Haixin Sun, and Guangjie Han

Abstract This chapter introduces underwater acoustic noise modeling based on generative-adversarial-network (GAN). In underwater acoustic communications, accurately fitting the impulsive noise is crucial. Traditional models with fixed parameters can only approximate the global heavy-tail distribution of the impulsive noise, failing to capture local distributions of varying lengths. To address this limitation, a GAN-based underwater noise simulator (GANUNS) has been presented. The GANUNS consists of a deep-neural-network-based generator and a convolutional-neural-network-based discriminator that learn the heavy-tail distribution of the impulsive noise. By utilizing real noise data collected in Xiamen, the simulated underwater acoustic noise generated by the GANUNS exhibits significantly lower Kullback–Leibler divergence, Jensen–Shannon divergence, and mean square error compared to traditional approximate models. This demonstrates the effectiveness of the suggested GANUNS in accurately modeling the impulsive noise for underwater acoustic communications.

J. Wang · M. Zhou (✉) · Y. Cui · H. Sun

Key Laboratory of Southeast Coast Marine Information Intelligent Perception and Application,
MNR, Zhangzhou, China

e-mail: mzzhou@xmu.edu.cn

H. Sun

e-mail: hxsun@xmu.edu.cn

J. Wang

Department of Communication Engineering, Tianjin University of Technology, Tianjin 300384,
China

M. Zhou · H. Sun

School of Informatics, Xiamen University, Xiamen 361001, China

Y. Cui

College of Computer and Information Engineering, Tianjin Normal University, Tianjin 300387,
China

G. Han

College of Internet of Things Engineering, Hohai University, Changzhou, China

1 Introduction

With the increasing requirements for the Internet of Things (IoT) in the oceans, efficient data transmission has become critical to ensure real-time underwater environment monitoring and emergency rescue. As underwater sensor networks and unmanned auto vehicles is being applied with large scales, the available underwater acoustic communication (UAC) frequency spectrum has been pushed and squeezed significantly [1, 2]. To address this problem, an efficient wireless modulation technique has been deployed into underwater acoustics [3–8]. Breaking through the bandwidth limitations of acoustic media and addressing the high-speed communication challenges in complex ocean acoustic channels are the primary research priorities in underwater communication today. In recent years, underwater multicarrier communication and its new regimes have gradually become the mainstream solution for future high-speed underwater communication [9]. However, in the study of Zhou et al., underwater multicarrier systems still need to seek spectrum efficiency improvement while ensuring low transmission errors due to the effects of underwater non-Gaussian noise, large delay-sparse multipath channels, and time-variant Doppler effects.

UAC systems are often disrupted by the impulsive interference which is non-Gaussian. This interference comes from various natural and man-made sources, such as ships' engines, animals, ice cracking, reflections from the seabed, sea waves and wind, and accidental hits in underwater acoustic signal transmissions. Impulsive interference can cause significant degradation in the performance of communication receivers and signal detectors, especially when traditional high-rate communication devices designed with Gaussian assumptions are utilized. The impulsive underwater ambient noise, which has been proven to be non-Gaussian, also poses challenges in evaluating underwater acoustic (UAC) systems [10–12]. This motivates scholars to study its characteristics because it makes system tests and evaluations less expensive and reproducible than field trials [13]. To address these problems, improved performance can be achieved by using statistical-physical models that are more appropriate for the impulsive interference.

The existing non-Gaussian models have fit the heavy-tail distribution of the impulsive underwater noise to different extent, such as the Middleton, Gaussian mixture (GM), and α -stable models. The GM model represents the non-Gaussian distribution as a mixture of independent Gaussian components with different mean and variance values. This model exhibits low precision in accurately fitting heavy tails of the impulsive underwater noise [14]. As the generalization of Gaussian distribution, α -stable model can represent various distributions (Gaussian, Poisson, Cauchy and other non-Gaussian distributions) with different parameters. It is often described by the integration of a characteristic function, where the α and β values control the number of impulses and symmetry of the distribution. Additionally, the Middleton Class A noise is an extensively studied probability distribution that effectively represents various physical phenomena. It is commonly employed to simulate narrowband impulsive noise in diverse systems. The Middleton Class A model can be interpreted

as a combination of independent impulsive source emissions, with the sources being distributed according to a Poisson distribution and their amplitudes following a Gaussian distribution. It should be noted that both α -stable and the Middleton model have no closed formulations [14, 15]. These parameter-fixed models can only fit the global distribution of the actual noise, with parameters that do not change over short duration [16–18]. It is challenging to learn the distribution of dynamic noise that changes in the short term within a specific area.

The generative adversarial network (GAN) is a promising approach to solving the above problem. It has been applied in noise reduction [19], signal classification [20, 21], and channel modeling [22–24]. The GAN learns the distributions with an unsupervised strategy, which has been applied in the end-to-end communication system [25]. The GAN is fed with random sequences to generate Rayleigh channels for training the communication network. Moreover, for specific distribution fitting, the GAN has been used to generate Gaussian and Rayleigh distributions [24]. The inspiring results show that applying the GAN for underwater impulsive noise fitting is feasible. Nevertheless, the aforementioned works indicate potential for optimizing the GAN-based model in real-world scenarios. This chapter will propose a GAN-based underwater noise-fitting strategy. A deep-neural-network (DNN)-based generator and a convolutional-neural-network (CNN)-based discriminator are built and utilized for real noise fitting. With the collected noise in the real sea area as training samples, the proposed GAN is trained and applied independently to generate the real noise, which can fit the real distribution of the noise with arbitrary length.

The remainder of this chapter is organized as follows. 2 analyses the underwater acoustic ambient noise collected in real sea environments. The fitting results using traditional non-Gaussian models are discussed and the problem of precision is highlighted. 3 proposes a GAN-based noise simulator and discusses its structure, 4 shows the experimental results of the proposed intelligent noise model, and 5 concludes this chapter.

2 Analysis of Real Underwater Noise

The actual underwater noise to be analyzed was collected using a single hydrophone deployed in the Xiamen sea area. Trials were conducted in August of the past several years to record noise signals in different positions, as shown in Fig. 1. The hydrophone was settled at a 4 m depth, and about 20 m far from the coast. The average depth of the seabed is 8 m. In addition to the natural noise from the friction between the seabed and the waters, the engines of fishing ships and sails are the main sources of ambient noise. Each noise sequence was collected at a length of 10 s. After filtering, the time domain and the spectrum of the sample noise are shown in Fig. 2. Though the spectrum is whitened, the impulses are found in the time domain signal.

As the collected noise contains too many points, a measured sequence should be segmented. Hence, k short slices are obtained from the long sequence of the actual noise represented as $[n_1(t) \cdots n_k(t)]$. To judge whether the real noise distribution



Fig. 1 Test spots in Wuyuanwan Bay

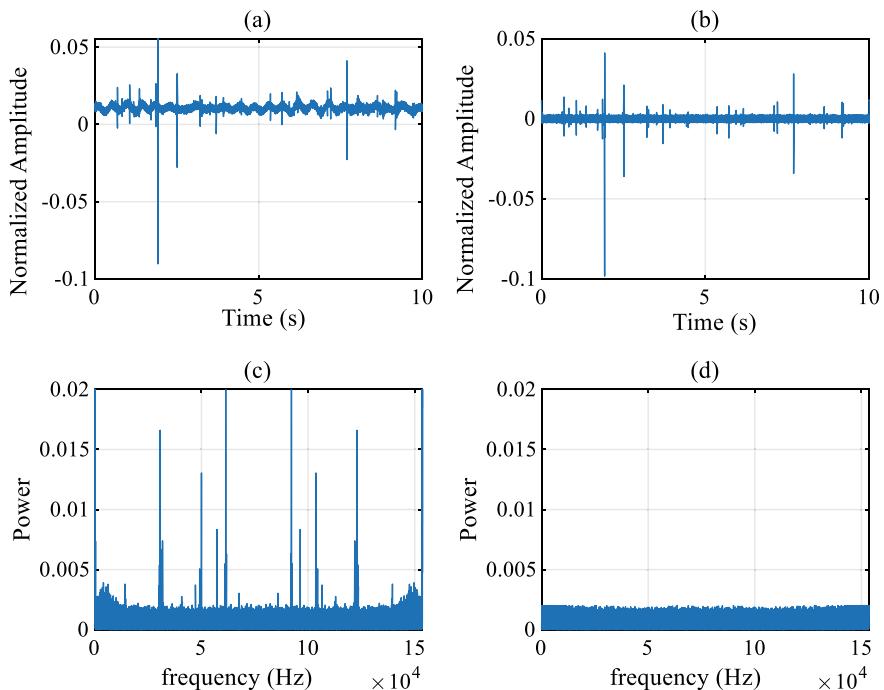


Fig. 2 The noise signals before and after filtering: **a** Time series of the original noise; **b** Time series of the noise without harmonical waves; **c** Spectrum of the original noise; **d** Spectrum of the noise without harmonical waves

is non-Gaussian, the Kolmogorov–Smirnov (K-S) test is conducted. The supremum of the cumulative density functions (CDF) of Gaussian $F_n(x)$ and noise slices $F(x)$ is [25]

$$D = \sup_x |F_n(x) - F(x)| \quad (1)$$

Thirteen real noise sequences were evaluated. The fiducial probability is set to 0.05; the probability of hypothesis rejection is shown in Fig. 3. Not all sequences reject the Gaussian hypothesis, especially at small lengths.

The above results show that the noise in the short length contains hybrid distributions. To fit them, the α -stable and GM models are taken into account. Therein, the probability distribution function (PDF) of the GM model can be expressed as

$$f(\omega) = \sum_{n=1}^N \varepsilon_n f_n(\omega), \quad (2)$$

where f_n is the n -th Gaussian component and ε_n is the corresponding coefficient. The number of Gaussian components N decides the complexity of the model [26], the typical value of which is 2.

The PDF of the α -stable model is usually represented as the integration of the character function

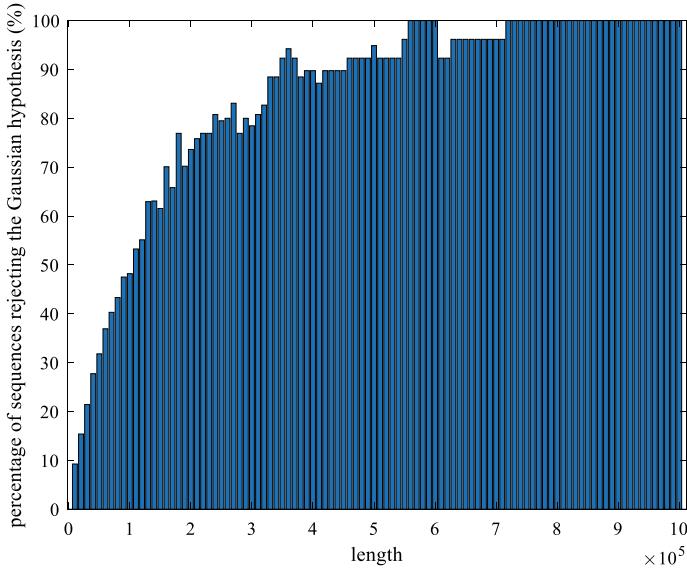


Fig. 3 Percentage of sequences rejecting the hypothesis with different slice lengths

$$f(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \varphi(t) e^{j\omega t} dt, \quad (3)$$

where

$$\varphi(t) = \begin{cases} \exp\left\{j\delta t - \gamma|t|^\alpha \left[1 + j\beta sign(t) \tan\left(\frac{\alpha\pi}{2}\right)\right]\right\}, & \text{if } \alpha \neq 1 \\ \exp\left\{j\delta t - \gamma|t|^\alpha \left[1 + j\beta sign(t) \frac{2}{\pi} \log|t|\right]\right\}, & \text{if } \alpha = 1 \end{cases}, \quad (4)$$

where α, β, γ and δ are factors affecting the thickness of tails, symmetry of the PDF, the dispersion from the mean, and the shift of the PDF, respectively. Equation (3) has no analytical representation but can be approximated by computer simulations [27, 28].

The expectation maximization (EM) algorithm is applied to fit the parameters of the GM and α -stable models. Table 1 shows the parameters fitted according to several different groups of real noise. Further, the PDFs of the GM model and α -stable model are computed and plotted, shown in Fig. 4.

Figure 4a shows the time series of the real and simulated noise, and PDFs of noise slices with a length of 10,000 were plotted in Fig. 4b. Figure 4a shows that the α -stable fitted noise contains too many impulses in the time domain, while the GM fitted noise contains almost no impulse. This phenomenon corresponds to Fig. 4b, where the PDF of a short length shows poor accuracy, especially on the tails of the PDFs. As a consequence, it can be concluded that the approximated models cannot well cover the local distributions of different lengths with fixed parameters, which leads to an inaccurate performance evaluation of communication systems on such noise. We apply GAN to learn the distributions with normalized inputs to obtain a more generalized model.

Table 1 Values of the fitted models according to the real noise

	Parameter	Group1	Group2	Group3
GM	ε_1	0.0526	0.0141	0.0109
	ε_2	0.9474	0.9859	0.9891
	μ_1	0.0045	-0.1534	-0.1322
	μ_2	0.0017	-0.0008	0.0021
	σ_1	192.6647	36.0198	17.3491
	σ_2	1.2188	0.8264	0.9081
α -stable	α	1.7875	1.9231	1.5210
	β	0.1054	0.0054	0.0149
	γ	0.0004	0.0004	0.0005
	δ	0	0	0

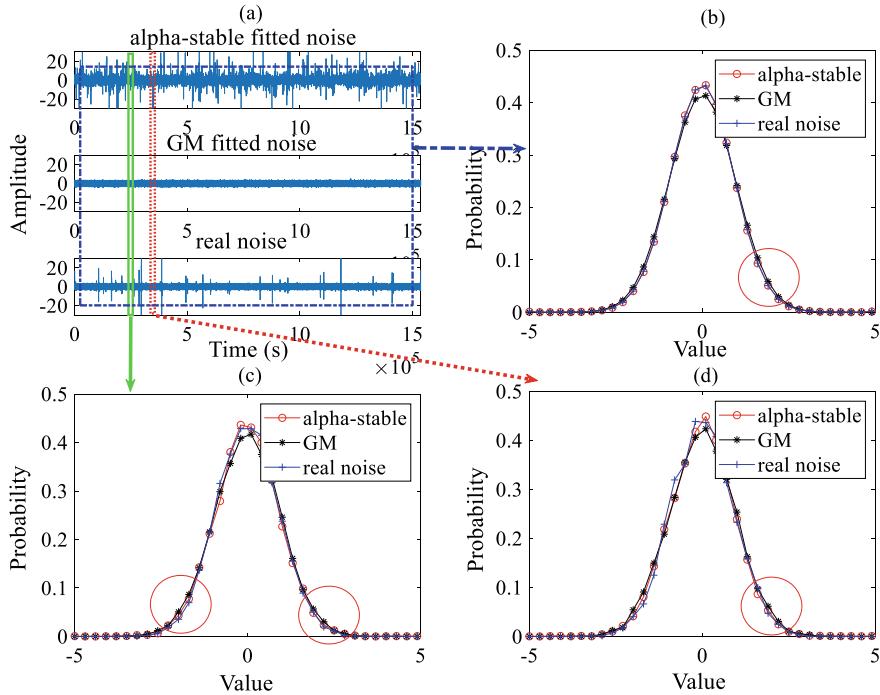


Fig. 4 Time series and corresponding real and simulated noise sequences PDFs

3 GAN-Based Underwater Noise Simulator

3.1 Introduction of GAN

According to Goodfellow's theory [29], for a multi-layer perceptron network, the learning of sequence distributions can be achieved using the framework of the GANs. A GAN typically consists of a generator and a discriminator, which are trained adversarially. The aim of generator is to learn the probability distribution $p_r(x)$ of the reference data x_r and generate independently distributed random variables as output. The following description will discuss the generator, discriminator, and training strategy with the underwater ambient noise as samples. To achieve this purpose, a noise variable is defined that follows a prior distribution $p_z(z)$, and then a generator function $G(z, \theta_G)$ is established by taking the noise sequence as input. This function represents a multi-layer perceptron with parameter set θ_G (referred to as $G(z)$), which maps the prior input z to the output sequence \hat{x} that fits the distribution. Additionally, another multi-layer perceptron network $D(x, \theta_D)$ (referred to as $D(x)$) is defined to discriminate the generator's output. θ_D represents the parameters set of this network, and $D(x)$ is the output discrimination index used to determine whether x comes from the generator's output distribution or the reference distribution $p_r(x)$.

For each generator output $G(z)$, the discriminator is firstly trained to maximize the probability of correctly classifying both the reference sequence x_r and $G(z)$. Since the generator's objective is to make the output sequence progressively closer to the distribution $p_r(x)$, the generator must confuse the discriminator between the generated sequence and the sample sequence, making it unable to distinguish between both types of sequences. This process can be modeled as a zero-sum game between the generator and the discriminator.

To construct an effective loss function, the GANs usually employ the Jensen-Shannon (JS) divergence for measuring the difference between the generator's output and the real distribution. It can be defined as follows

$$JS[p_r, p_g] = \frac{1}{2}KL\left[p_r \parallel \frac{p_r + p_g}{2}\right] + \frac{1}{2}KL\left[p_g \parallel \frac{p_r + p_g}{2}\right], \quad (5)$$

where p_g represents the probability distribution of the generated sequence, and $KL(\cdot)$ denotes the Kullback–Leibler (KL) divergence. For two probability distributions p_a and p_b , their KL divergence can be expressed as

$$KL(p_a \parallel p_b) = \mu \int \log\left(\frac{p_a(x)}{p_b(x)}\right) p_a(x) dx \quad (6)$$

where μ represents the mean of the probability distribution p_a . Based on the above representation, the optimization objective of GAN can be expressed as

$$\min_G \max_D V(D, G) = E_{x \sim p(x)}[\log D(x)] + E_{z \sim p(z)}[\log(1 - D(G(z)))] \quad (7)$$

The GANs obtain the statistical distribution from the generator network, and the learned data distribution is updated through the gradients from the discriminator, thus the distribution fitting problem is transformed into a backpropagation problem. For the sample sequence, the generator provides a random output for each random prior input z . As mentioned in the last section, the underwater environmental noise is a combination of Gaussian distribution and non-Gaussian distribution at a small scale, the distributions in different time slices are characterized by uncertainty. Compared to α -stable distributions and Gaussian mixture models, the GANs can comprehensively cover the values of various scale slices when dealing with such complex distributions, theoretically allowing for a better restoration of the sequence distribution. Therefore, the following will focus on fitting the distribution of underwater environmental noise and experimentally validate the effectiveness of the proposed network through fitting the real measured noise.

3.2 GAN Structure for Noise Simulations

As shown in Fig. 5, a generator is built to output imitative noise with random inputs, which can be treated as a randomizer. For each input value, there is a random sequence to satisfy the $n - \text{to} - 1$ mapping, which is written as

$$\hat{\mathbf{n}} = G(\mathbf{s}), \quad (8)$$

where $\hat{\mathbf{n}}$ is an $N_{\text{BG}} \times 1$ vector and \mathbf{s} is an $N_{\text{BG}} \times k_g$ matrix. Both variables contain N_{BG} batches. Furthermore, k_g is small in most situations. Consequently, the fully-connected hidden layers model is sufficient to deal with the random inputs.

Unlike the generator, the goal of the discriminator is to evaluate how much a sequence is “real”. For these sequences, the CNN is applied to extract features of the reshaped sequences. The function of the discriminator can be written as

$$\mathbf{I} = D(\mathbf{n}), \quad (9)$$

where \mathbf{I} is an $N_{\text{BD}} \times 1$ vector representing the indexes of the input sequence containing N_{BD} batches. \mathbf{n} is the input noise matrix, whose shape is $N_{\text{BD}} \times k_d$. N_{BD} , N_{BG} and k_d should satisfy

$$N_{\text{BG}} = N_{\text{BD}} \times k_d. \quad (10)$$

Assuming I_R and I_G to be the distance between the outputs of real and fake samples; in training, these two values should increase with epoch. Hence, the loss

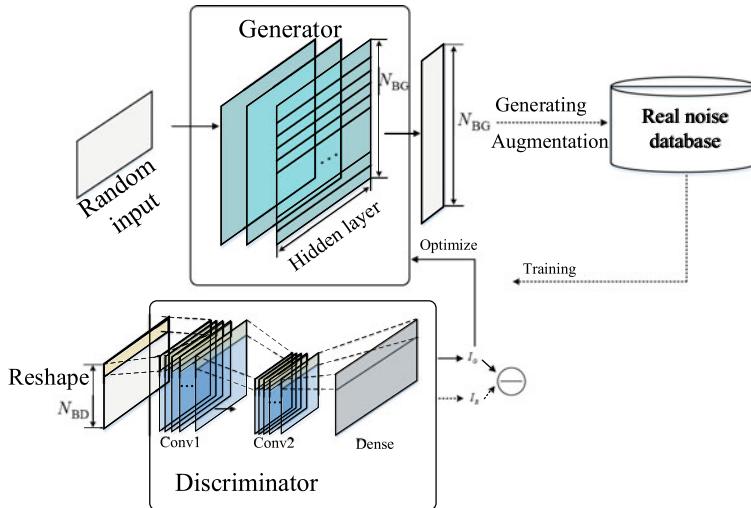


Fig. 5 Structure of the proposed GAN

function of the discriminator is represented as

$$L_D = I_G - I_R. \quad (11)$$

As mentioned earlier, a zero-sum game is built between the generator and the discriminator. The loss function of the generator should correspond to that of the discriminator, yielding

$$L_G = -I_G. \quad (12)$$

The value of L_D and L_G denotes the distance between the output of the discriminator with real and generated samples.

In each epoch of training, to obtain the stable judgment of the generated and real noise, the discriminator should be trained with more iterations than the generator. As shown in Algorithm 1, in each epoch, $\text{opt}(\cdot)$ represents the optimization process of the network. The discriminator is trained with additional iterations, while the generator is iterated once. Subsequently, the generator continues converging until the loss function L_D approaches zero. It denotes that the discriminator cannot distinguish the generated and actual noise sequences.

Algorithm 1 Optimization of One Epoch in the Training of the GAN

Input Collected noise \mathbf{N} from real scenarios

```

Initialize  $G(\cdot), D(\cdot), N_e, iter = 0, flag = 0, Th_d$ 
1. while  $iter < N_e$ .
2.      $\mathbf{n} = \text{rand}(N_{BG}, k_g);$ 
3.      $\hat{D}(\mathbf{n}) = \text{opt}(D(\mathbf{n}));$ 
4.      $iter = iter + 1;$ 
5.      $L_D = D(\mathbf{n}) - D(\mathbf{N});$ 
6.     if  $L_D < Th_d$ .
7.          $flag = 1;$ 
8.         break;
9.     end if
10.    if  $flag = 1$ .
11.        Return  $\hat{D}(\cdot), \hat{G}(\cdot);$ 
12.    end if.
13.     $\mathbf{n} = \text{rand}(N_{BG}, k_g);$ 
14.     $\hat{G}(\mathbf{n}) = \text{opt}(G(\mathbf{n}));$ 
15.    Return  $\hat{D}(\cdot), \hat{G}(\cdot).$ 
```

4 Numerical Simulations and Discussions

4.1 Parametric Configuration of GAN-Based Noise Simulator

To analyze the performance of the presented GAN-based noise simulator, the real noise is taken as the sample to train the GAN. Same as the configuration in Sect. 2, 13 noise sequences are applied, and each length is 1536000. The parameters of the network are listed in Table 2. A generator with three dense layers is trained jointly with a CNN-based discriminator. The batch sizes of the generator and discriminator are 12,000 and 30, respectively. The kernel size of each convolutional layer is 2, and two pooling layers are applied for under-sampling. Moreover, all the networks are activated by the leaky rectified linear unit (Leaky ReLU) function.

The GAN is trained directly with the parameters listed in Table 3, where the learning rate is set to be 0.00001. Figure 6 shows the loss function of the discriminator while training. A valley is observed around the 570th epoch, denoting that the training of the discriminator is finished. As the generator continues to improve, the loss function becomes smaller and smaller, eventually converging to a near-zero value. This indicates that the discriminator cannot differentiate between the actual noise and the generated noise.

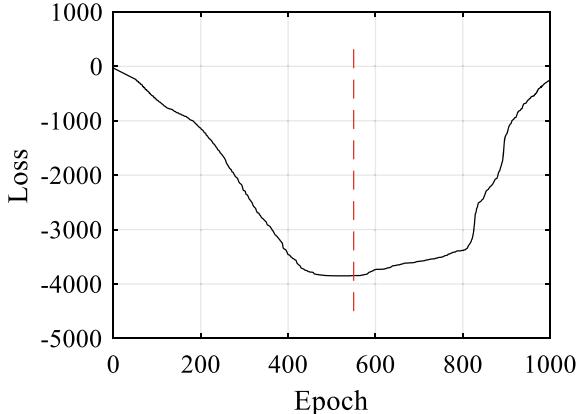
Table 2 Parameters of the proposed GAN

Generator	
Layer	Dimension
Input	128
Dense 1	100
Dense 2	100
Dense 3	100
Output	1
Discriminator	
Input	400
Conv 1	14 (Channel)
Pooling	
Conv 2	14 (Channel)
Pooling	
Flatten	
Output	1

Table 3 Parameters of training

	Generator training	Discriminator training
Optimizer	Adam	Adam
Learning rate	0.00001	0.00001

Fig. 6 Converging curve of the GAN



4.2 Accuracy Analysis for GAN and Traditional Non-gaussian Models

The trained generator is taken to generate the noise sequences that will be further compared with the GM and α -stable models. As shown in Fig. 7, the global PDFs of different sequences with a length of 1,536,000 are plotted together. Figure 7b shows the local PDFs with a length of 10,000. The probability in the range of [2.2, 3] is zoomed in. It can be seen from both figures that the probabilities of the GAN almost coincide with that of the actual noise.

The quantification of the different PDFs above should be compared with the possibilities between the real and generated noise. The mean square errors (MSE), Kullback–Leibler (KL) divergences, and Jensen–Shannon (JS) divergences are considered as the indicators [17]. Different lengths are chosen to segment the noise sequences, the MSEs are firstly shown in Fig. 4.8. The GAN reaches the lowest MSE with all lengths, while the α -stable model shows the worst performance. The stability of the three models deteriorates for shorter lengths due to increased randomness.

Similar results can be observed in the KL divergences and JS divergences as depicted in Figs. 9 and 10. The GAN always shows the highest precision compared with the other two models. Another phenomenon can be observed that despite the higher accuracy of the α -stable model than the GM model with long sequences, the α -stable model shows the worst performance at short lengths. From Fig. 4, it can be noticed that the α -stable model generates many impulses. Almost all the generated sequences by the α -stable model reject the Gaussian hypothesis. The GAN can balance the conflicts of the noise lengths and reach the highest accuracy.

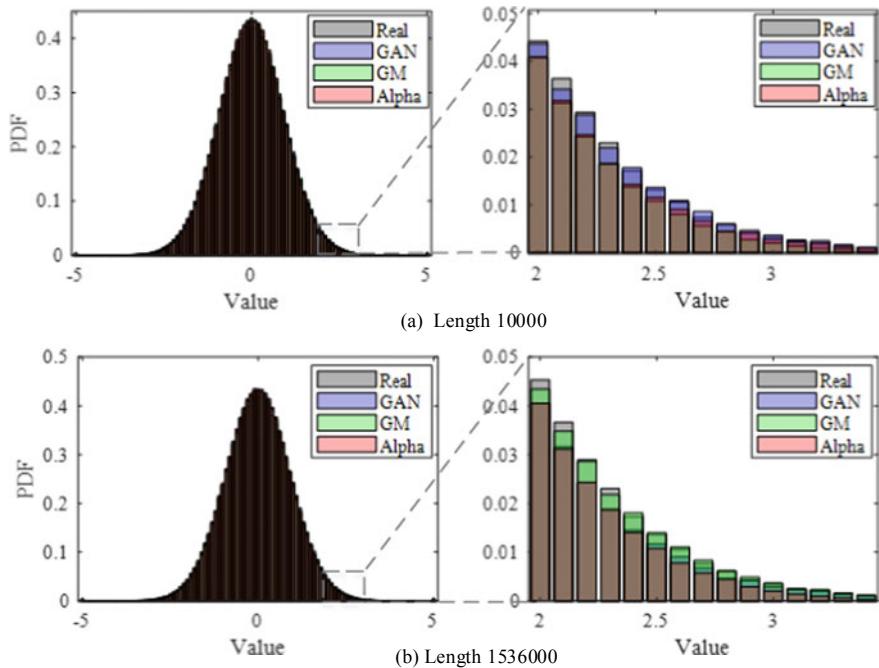


Fig. 7 PDFs of different sequences. **a** The global PDFs of different models with a length of 1,536,000. **b** The local PDFs with a length of 10,000

The numerical results above show the stability and accuracy of the suggested GAN-based simulator in underwater noise fitting. For traditional communication systems, more reliable simulated noise can help to estimate the performance without large amounts of trials precisely, and thus the specific optimization of the system becomes easier to be conducted for adaptation to the target sea environment. Also, reliable noise samples are determinative to the training of the deep-neural-network (DNN)-based communication systems. With a large number of simulated noise samples, the DNN-based receiver will be trained to learn the noise distribution of real environments to the best of its ability.

To show the effect of the GAN on the receiver, we test the BER of the BPSK modulation with different kinds of noise, as shown in Fig. 11. The real noise collected in the real sea is applied for performance evaluation. The GAN, GM, and α -stable models are used to generate the noise with estimated distribution from the real noise. It can be seen that the BER of the GM model starts to deviate from that of the real noise when $SNR > 2$ dB, while the α -stable model shows inaccuracy when $SNR > 9$ dB. Only the GAN is always consistent with the real noise in each value of SNR.

4.3 Complexity Analysis

Further, to clarify the effectiveness of the suggested GAN, the computational complexity for different models is discussed, along with the tradeoffs between the performance and complexity of the GAN. It should be noted that the GAN presented in this work does not serve real-time systems. Similar to the GM and α -stable models, the model training and application of the GAN are separated. In the training process, the generator and the discriminator are trained alternately. The complexity of the GAN comes from the hidden dense layer. Given that the number of neurons J in each layer is the same, the computational complexity of the generator is

$$Time_G \sim O\left(B_G \sum_{l=1}^{N_G} [(2J_{l-1} - 1)J_l]\right) \sim O(B_G J^2). \quad (13)$$

where N_G is the number of hidden layers, which is much smaller than J . B_G is the batch size of the generator.

The complexity of the discriminator is from convolutional layer (CL) and the output layer. The number of the CLs N_D , the kernel size K_l and the number of channels C_l of the l -th layer are much smaller than the input dimension. Therefore, the computation complexity of the discriminator is

$$Time_D \sim O\left(B_D \sum_{l=1}^{N_D} [M_l K_l C_{l-1} C_l] + J_F\right) \sim O(B_D M + B_D J_F), \quad (14)$$

where J_F is the dimension of the flattened layer and B_D is the batch size of the discriminator.

For the GM model, the complexity of parameter estimation with the expectation maximization (EM) algorithm is

$$Time_{GM} \sim O(N_{Gaussian} K_P), \quad (15)$$

where K_P is the input length for parameter estimation and $N_{Gaussian}$ is the number of multiply-accumulate calculations (MACC) in the EM algorithm of the GM model.

For α -stable model, the complexity of the maximum-likelihood estimation (MLE) algorithm is

$$Time_\alpha \sim O(N_\alpha K_P), \quad (16)$$

where N_α is the number of multiply-accumulate calculations (MACC) in the MLE algorithm of α -stable model. The above results show that the GAN requires a large amount of training time in comparison to the GM and α -stable models. However, the GAN outperforms the GM and alpha-stable models to a large extent in precision. From Fig. 8, 9 and 10, the numerical results show a leap of the precision of the GAN

compared with the other two models, particularly at short lengths. This means the generated noise by the GAN is much more reliable than the GM and alpha stable models as the extended noise samples for the actual target sea area.

Moreover, the above complexity of the GAN has the same order as [30], which is acceptable for communication systems. In application, the generator can be applied singularly for noise simulation after being trained, further reducing the computations. The GAN with higher precision is very helpful for sample extension of the DNN-based communication systems. Hence, the GAN is worthwhile to pay more computation for higher model precision. In data simulations, only the generator of the GAN is used for noise simulation. Hence, the complexity can be represented as follows.

$$Time_G \sim O(K_G J^2). \quad (17)$$

Fig. 8 MSEs of different PDFs

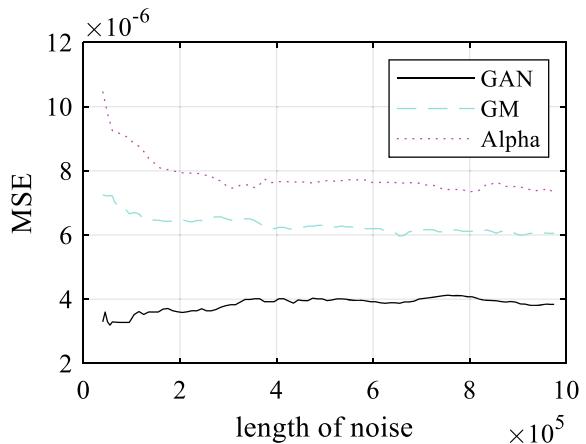


Fig. 9 KL divergences of different PDFs

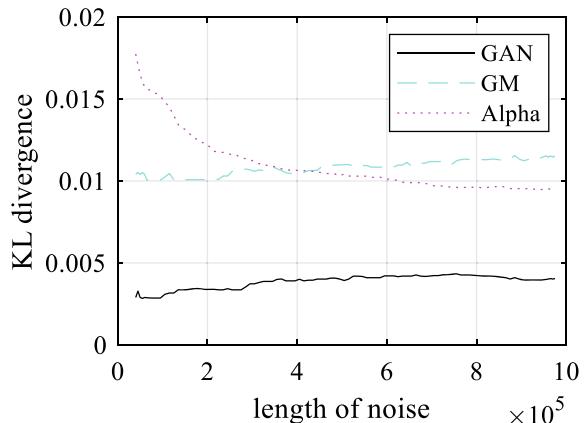


Fig. 10 JS divergences of different PDFs

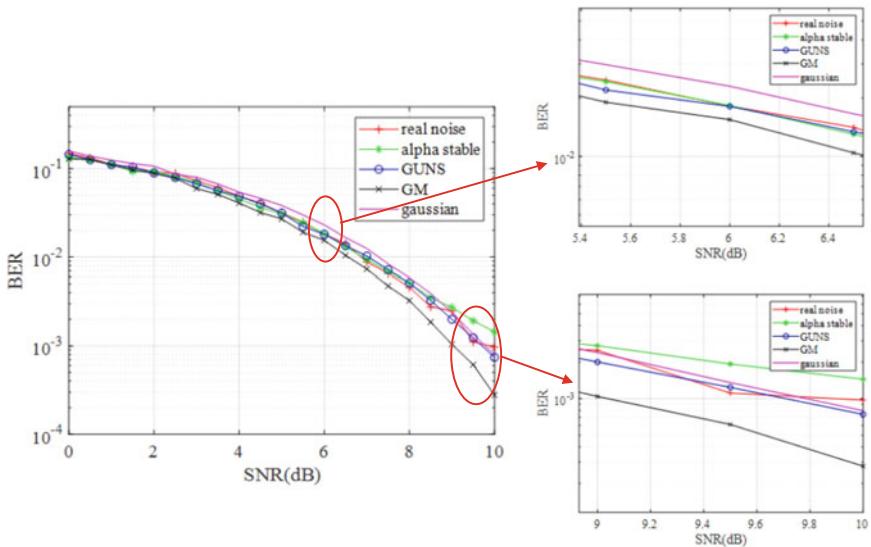
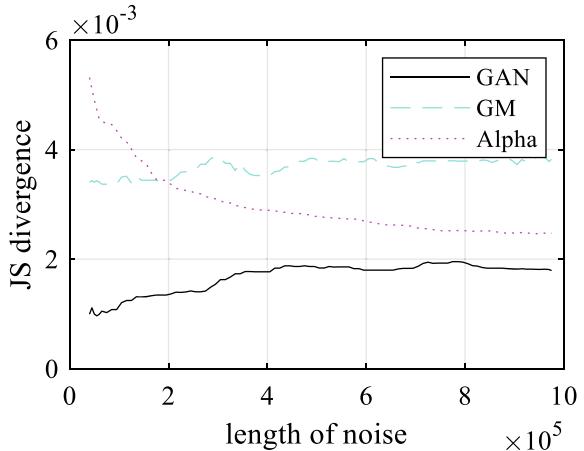


Fig. 11 BER of BPSK with different types of noise

Since GM generator and the computer-simulated α -stable generator have fixed parameters, the complexity of the two generators can be simplified as

$$Time_{\text{GMgen}} \sim O(N_{\text{Gaussian}} K_G), \quad (18)$$

$$Time_{\alpha\text{gen}} \sim O(N_\alpha K_G), \quad (19)$$

Apparently, $J^2 > N_{\text{Gaussian}}$ and $J^2 > N_\alpha$. Despite this, in this work, a small network with $J = 100$ is utilized, which is easy to be implanted into the hardware as the DSP chips reach high speed today. In our future work, more scientific choices of parameters will be made for different environments with different noise distributions.

What's more, the GAN is to be further optimized in the future for higher accuracy, better generalization ability and lower complexity.

5 Conclusions

In this chapter, a GAN-based underwater impulsive noise simulator have been proposed. A simple structure of the DNN and the CNN with fewer parameters is designed for the training. Compared with the GM and α -stable models, the suggested GAN reaches the lowest MSE, KL divergence, and JS divergence on both local and global distributions. Trained with collected actual noise, the addressed GAN can generate reliable samples for optimization and evaluation of the performance of underwater acoustic communication systems. The future works should be focused on the adaptation of the network in the changing environments and the optimization of the precision and complexity of the network structure.

Acknowledgements This work was supported in part by Key Laboratory of Southeast Coast Marine Information Intelligent Perception and Application, MNR, NO. 220202, as well as in part by National Natural Science Foundation of China under Grants 62271426 and 62201385.

References

1. Wang, Q., Dai, H., Cheang, C. F., & Wang, H. (2017). Link connectivity and coverage of underwater cognitive acoustic networks under spectrum constraint. *Sensors*, 17(12), 2839.
2. Moreno-Roldán, J. M., Poncela, J., Otero, P., et al. (2020). A no-reference video quality assessment model for underwater networks. *IEEE Journal of Oceanic Engineering*, 45(1), 342–346.
3. Chen, Z., Wang, J., & Zheng, Y. R. (2017). Frequency-domain turbo equalization with iterative channel estimation for MIMO underwater acoustic communications. *IEEE Journal of Oceanic Engineering*, 42(3), 711–721.
4. Wang, J., Cui, Y., Li, J., et al. (2020). On orthogonal coding-based modulation. *IEEE Communications Letters*, 24(4), 816–820.
5. Tao, J. (2018). DFT-precoded MIMO OFDM underwater acoustic communications. *IEEE Journal of Oceanic Engineering*, 43(3), 805–819.
6. Bocus, M. J., Doufexi, A., & Agrafiotis, D. (2020). Performance of OFDM-based massive MIMO OTFS systems for underwater acoustic communication. *IET Communications*, 14(4), 588–593.
7. Emokpae, L. E., Freeman, S. E., Edelmann, G. F., et al. (2019). Highly directional multipath free high data-rate communications with a reconfigurable modem. *IEEE Journal of Oceanic Engineering*, 44(1), 229–239.

8. Wen, M., Cheng, X., Yang, L., et al. (2016). Index modulated OFDM for underwater acoustic communications. *IEEE Communications Magazine*, 54(5), 132–137.
9. Zhou, S., & Wang, Z. (2014). OFDM for underwater acoustic communications. Wiley.
10. Kuai, X., Sun, H., Zhou, S., et al. (2016). Impulsive noise mitigation in underwater acoustic OFDM systems. *IEEE Transactions on Vehicular Technology*, 65(10), 8190–8202.
11. Wang, J., Li, J., Yan, S., et al. (2021). A novel underwater acoustic signal denoising algorithm for GAUSSIAN/non-GAUSSIAN impulsive noise. *IEEE Transactions on Vehicular Technology*, 70(1), 429–445.
12. Liu, S., Xiao, L., Huang, L., et al. (2019). Impulsive noise recovery and elimination: A sparse machine learning based approach. *IEEE Transactions on Vehicular Technology*, 68(3), 2306–2315.
13. Banerjee, S., & Agrawal, M. (2014). On the performance of underwater communication system in noise with Gaussian mixture statistics. In Paper presented at the 20th National Conference on Communications (NCC). Kanpur, India.
14. Banerjee, S., & Agrawal, M. (2013). Underwater acoustic noise with generalized gaussian statistics: Effects on error performance. IN Paper presented at 2013 MTS/IEEE OCEANS, Bergen, Norway.
15. Lin, B., Wang, X., Yuan, W., et al. (2020). A novel OFDM autoencoder featuring CNN-based channel estimation for internet of vessels. *IEEE Internet of Things Journal*, 7(8), 7601–7611.
16. Zhang, X., Ying, W., & Yang, B. (2018). Parameter estimation for class a modeled ocean ambient noise. *Journal of Engineering and Technological Sciences*, 50(3), 330–345.
17. Panaro, J. S. G., Lopes, F. R. B., & Barreira, L. M., et al. (2012). Underwater acoustic noise model for shallow water communications. In Paper presented in Brazilian telecommunication symposium. Brasilia, Brazil.
18. Agrawal, A., Kumar, R., & Agrawal, M. (2019). Modeling of underwater noise. Paper presented in OCEANS 2019. Marseille, France.
19. Pascual, S., Park, M., & Serrà, J. et al. (2018). Language and noise transfer in speech enhancement generative adversarial network. In Paper presented in 2018 IEEE inter-national conference on acoustics, speech and signal processing (ICASSP), Calgary, AB, Canada.
20. Ji, X., Wang, J., Li, Y., et al. (2020). Data-limited modulation classification with a CVAE-enhanced learning model. *IEEE Communications Letters*, 24(10), 2191–2195.
21. Yao, X., Yang, H., & Li, Y. (2019). Modulation identification of underwater acoustic communications signals based on generative adversarial networks. In Paper presented in OCEANS 2019. Marseille, France.
22. Tolba, B., Elsabrouty, M., Abdu-Aguye, M. G., et al. (2020). Massive MIMO CSI feedback based on generative adversarial network". *IEEE Communications Letters*, 24(12), 2805–2808.
23. Dong, Y., Wang, H., & Yao, Y. D. (2020). Channel estimation for one-bit multiuser massive MIMO using conditional GAN. *IEEE Communications Letters*, 25(3), 854–858.
24. Yang, Y., Li, Y., Zhang, W., et al. (2019). Generative-adversarial-network-based wireless channel modeling: Challenges and opportunities. *IEEE Communications Magazine*, 57(3), 22–27.
25. Ye, H., Li, G. Y., & Juang, B. H. F., et al. (2018). Channel agnostic end-to-end learning-based communication systems with conditional GAN. In Paper presented in 2018 IEEE Globecom Workshops (GC Wkshps). Abu Dhabi, United Arab Emirates.
26. Massey, F. J. (1951). The Kolmogorov-Smirnov test for goodness of fit. *Publications of the American Statistical Association*, 46(253), 68–78.
27. Gopal, S. S. (2002). Finite mixture models. Advanced algorithmic approaches to medical image segmentation: State-of-the-art applications in cardiology, neurology, mammography and pathology, pp 341–361

28. Weron, A., Weron, R. (1995). Computer simulation of Le'vy α -stable variables and processes. In Paper presented in Chaos—The Interplay between stochastic and deterministic behaviour: Proceedings of the XXXIst winter school of theoretical physics held. Karpacz, Poland.
29. Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al. (2020). Generative adversarial nets. *Communications of the ACM*, 63(11), 139–144.
30. Lee-Leon, A., Yuen, C., & Herremans, D. (2021). Underwater acoustic communication receiver using deep belief network. *IEEE Transactions on Communications*, 69(6), 3698–3708.

How Generative AI Is Transforming Medical Imaging: A Practical Guide



Khaled ELKarazle, Valliappan Raman, Patrick Then, and Caslon Chua

Abstract Medical imaging is a crucial aspect of modern healthcare, as it enables the diagnosis and treatment of various diseases and conditions. However, developing and deploying AI models for medical imaging is challenging, due to the limited availability and quality of data, as well as the high complexity and diversity of imaging modalities and tasks. Generative AI models, such as variational autoencoders (VAEs), generative adversarial networks (GANs), and text-to-image diffusion models, offer a promising solution to these challenges, as they can generate realistic and diverse images from existing data or latent representations. In this chapter, we provide a practical guide on how generative AI is transforming medical imaging, by reviewing the state-of-the-art methods and frameworks, presenting some successful case studies in different domains and modalities, and discussing the future directions and opportunities for research and development.

1 Generative AI for Images: What It Is and How It Works

Generative AI is a branch of artificial intelligence that aims to generate novel content or modify existing content based on certain criteria or constraints, using various computational techniques and algorithms as defined by [20, 27]. Unlike conventional predictive AI models that aim to produce a prediction based on a given input, generative AI models focus on learning how to create or modify content according

K. ELKarazle (✉) · P. Then

Faculty of Engineering, Computing, and Science, Swinburne University of Technology Sarawak Campus, 93350 Kuching, Sarawak, Malaysia
e-mail: kelkaeazle@swinburne.edu.my

V. Raman

Department of Artificial Intelligence and Data Science, Coimbatore Institute of Technology, Coimbatore, Tamil Nadu 641014, India

C. Chua

Faculty of Science, Engineering and Technology, Swinburne University of Technology, Melbourne, VIC 3122, Australia

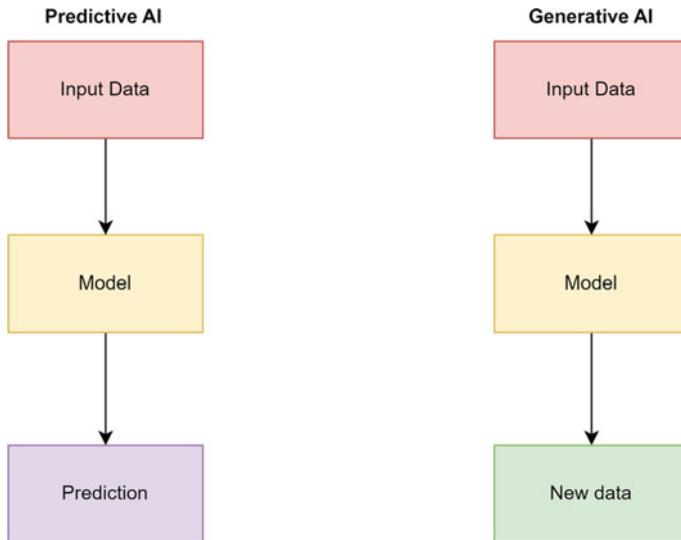


Fig. 1 An overview of the difference between predictive AI and generative AI

to certain criteria or constraints. In Fig. 1, we depict a comparison between the processing of data by generative AI and predictive AI.

Generative AI models encompass a diverse range of architectures and are constructed based on distinct underlying philosophies. Generative AI has been employed in a diverse array of tasks, including the generation and augmentation of images, videos, and audio. More recently, it has been utilized in the generation of text through the implementation of large language models such as the Generative Pre-trained Transformer (GPT), which has shown impressive results in various tasks.

2 GANs, VAEs and Diffusion Models: What Are They?

In the field of artificial intelligence, there exists a multitude of generative models that are capable of generating, enhancing, and synthesizing images. Among the most prominent architectures that have been widely utilized in addressing computer vision problems are generative adversarial networks (GANs), variational autoencoders (VAEs), and diffusion models. These models have demonstrated remarkable performance in generating high-quality images that are often indistinguishable from real photographs.

Generative Adversarial Networks (GANs) were initially introduced in 2014 by [7] and since their inception, they have undergone significant evolution, with the introduction of new models over the years that are capable of generating instances that are virtually indistinguishable from real data.

Generative Adversarial Networks (GANs) employ a min–max game framework, wherein two neural networks are trained with opposing objectives, and each network facilitates the enhancement of the other’s output. A prototypical GAN comprises a generator network that ingests a latent vector and endeavors to synthesize novel content that is sufficiently realistic to deceive the other network, designated as the discriminator. The discriminator receives inputs from both the generator’s output and real samples. Its objective is to determine whether the input sample originated from the actual dataset or was produced by the generator. Over several epochs, the discriminator gets better in detecting artificially generated samples while the generator improves the quality of the generated samples. Figure 2 illustrates a summary of how a GAN network is trained to generate novel samples.

Numerous variations of GANs have been developed, each with its own unique philosophy and designed to address specific challenges. Despite the diversity in their design and objectives, all GANs share a common training philosophy. Some of the most widely used types of GANs include Vanilla GANs, Conditional GANs (CGANs), Deep Convolutional GANs (DCGANs), CycleGANs, Generative Adversarial Text to Image Synthesis, StyleGANs, and Super Resolution GANs (SRGANs). Table 1 presents more details about each type of GAN.

Variational Autoencoders (VAEs) are a distinct class of generative artificial intelligence models that operate using a slightly different mechanism compared to Generative Adversarial Networks (GANs). While both VAEs and GANs are capable of generating new data samples, they employ different architectures and training methods to achieve this goal.

VAEs consist of an encoder and a decoder module, while GANs consist of a generator and a discriminator module. During training, VAEs learn the probability distribution that models the input data, while GANs learn through a competitive

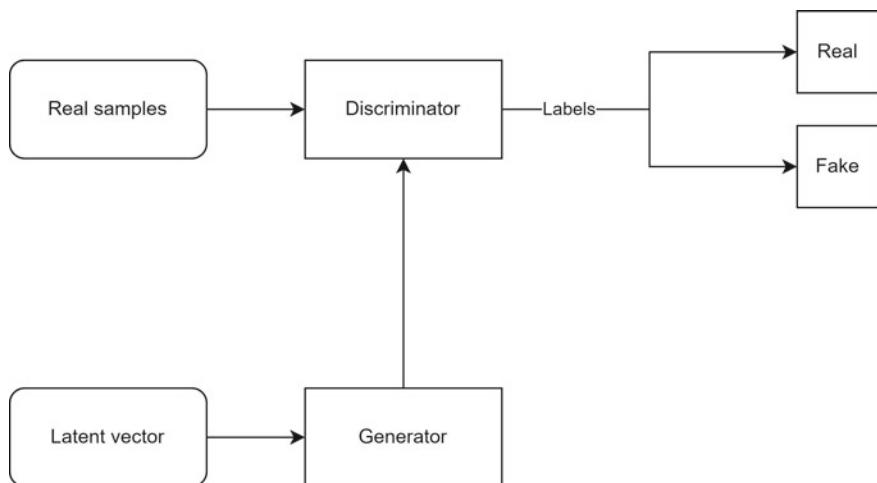


Fig. 2 The training process of a typical GAN network

Table 1 The various types of GANs and their common usage

GAN type	Description
Vanilla GAN	This is the simplest type of GAN, made up of a generator and a discriminator. It is used for the classification and generation of images
Conditional GAN (CGAN) [12]	In this type of GAN, both the generator and discriminator are provided with additional information, such as a class label or other data. This additional information helps the discriminator determine the conditional probability instead of the joint probability
Deep convolutional GAN (DCGAN) [15]	This type of GAN uses a deep convolutional network in its generator, allowing it to generate high-resolution and high-quality images
CycleGAN	This type of GAN is used for image-to-image translation, where one image is mapped to another image
Generative adversarial text to image synthesis [16]	This type of GAN is used to generate images from text descriptions
StyleGAN [8]	This type of GAN is used to generate images with a specific style or aesthetic
Super resolution GAN (SRGAN)	This type of GAN is used to increase the resolution of images

process involving the generator and discriminator modules. A typical VAE consist of two main components: an encoder module and a decoder module. The encoder takes an input image or video and maps it to two parameters in a latent space of representations: z_{mean} and $z_{log(variance)}$.

A point z is then randomly sampled from the latent normal distribution that is assumed to generate the input image or video, via $z = z_{mean} + (z_{log(variance)})^e \times \varepsilon$, where ε is a random tensor of small values. The sampled point z is then fed into the decoder module, which uses it to reconstruct the original image or video. During the training process, Variational Autoencoders (VAEs) are designed to learn the underlying probability distribution of the input data. Once the training is complete, the generative model can generate new, realistic samples by drawing points from the learned distribution and passing them through the decoder module to produce the final output. Figure 3 illustrates a typical architecture of a VAE.

Text-to-image diffusion models are a type of generative AI model that can generate images from text descriptions. One example of such a model is Imagen, a text-to-image diffusion model developed by Google Research.

Imagen builds on the power of large transformer language models in understanding text and hinges on the strength of diffusion models in high-fidelity image generation. The model uses a large frozen T5-XXL encoder to encode the input text into embeddings. A conditional diffusion model then maps the text embedding into a 64×64 image.

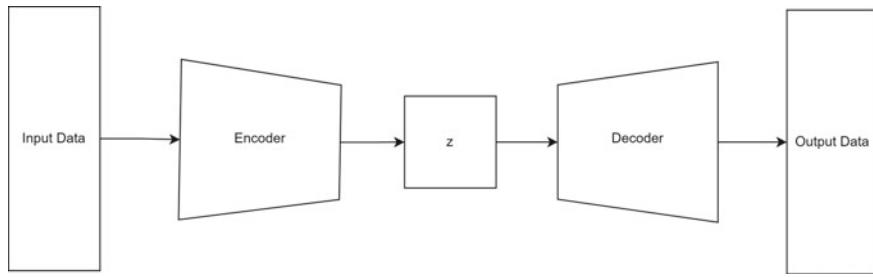


Fig. 3 The design of a typical Variational autoencoder

Diffusion models have emerged as a popular approach for a wide range of generative tasks, including text-to-image generation. These models work by reversing the process of adding noise to data, gradually denoising it to produce a final sample.

One key difference between text-to-image diffusion models and GANs/VAEs is that diffusion models create samples step by step, first creating a coarse image structure and then focusing on adding fine details on top. This gradual process allows diffusion models to produce high-fidelity samples. Figure 4 demonstrates the capability of diffusion models to produce high-quality medical images.

In contrast, Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) generate samples at once. GANs consist of a generator and a discriminator module that learn through a competitive process during training. VAEs consist of an encoder and a decoder module that learn the probability distribution that models the input data during training. The key differences between the three architectures are presented in Table 2.

Table 2 A comparison of the key differences between the three generative AI architectures

Property	GANs	VAEs	Text-to-image diffusion
Design	Consists of a generator and discriminator models	Consists of an encoder and decoder modules	Uses large transformer language model combined with conditional diffusion model
Training process	Learns in a competitive process between the generator and the discriminator	Uses probability distribution that models the input data	Works by reversing the process of adding noise to data, gradually denoising it to produce a final sample
Sample creation	Samples are generated at once	Samples are generated at once	Samples are created gradually which allows the creation of high-fidelity samples

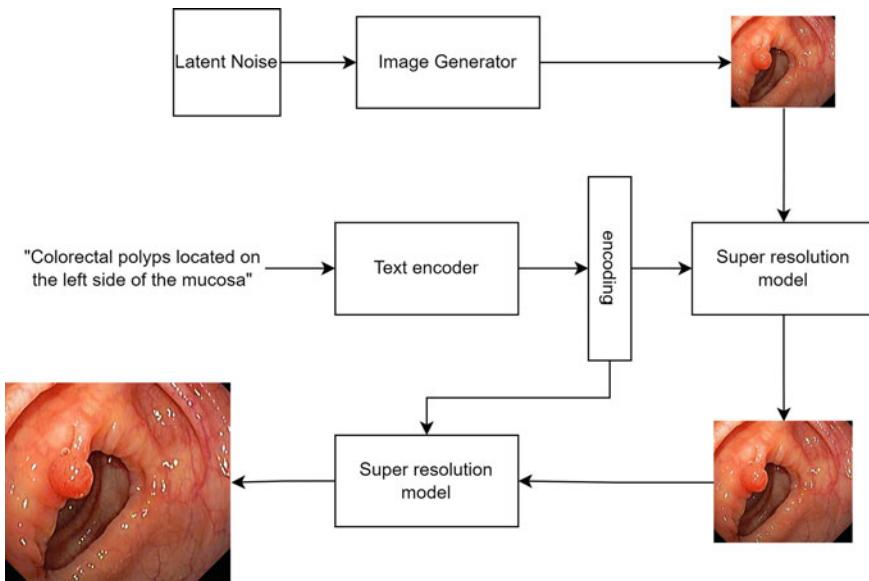


Fig. 4 An overall design of a typical text-to-image diffusion model. The image used in this illustration is taken from a public dataset

3 How Can Generative AI Help Medical Imaging?

In light of the burgeoning interest in the digitalization of the healthcare sector, researchers have been developing an array of AI tools over the years that function as assistive instruments to alleviate the stress and fatigue experienced by clinicians. Examples of such tools include object detection and segmentation systems that are capable of identifying various objects in endoscopy images [22] or the detection of abnormalities in brain Magnetic Resonance Imaging (MRIs) [6], chest Computed Tomography (CT) scans [25], and X-rays [4]. These AI-powered systems can assist clinicians in making more accurate diagnoses and developing more effective treatment plans.

Nonetheless, as with any other AI system, the acquisition of a sufficient quantity of training samples that encompass the majority of possible examples presents a significant challenge. This is because the effectiveness and generalization of those systems is largely dependent on the quality and quantity of the data used to train it.

One approach to addressing the issue of generalization is to augment the quantity of training samples by acquiring additional images from patients. While this may partially mitigate the problem of generalization, it raises a host of ethical and privacy concerns. An alternative method for working with a limited dataset is to employ data augmentation techniques, which have demonstrated efficacy in enhancing the performance of AI tools.

While data augmentation can serve as an efficacious approach for augmenting the quantity of training data, it does not inherently address the issue of generalization. For example, an individual may manifest colorectal polyps that diverge substantially from those represented in a given dataset, and an AI system trained exclusively on that dataset may be incapable of accurately identifying or diagnosing the condition.

In an effort to address the aforementioned challenges, AI researchers have turned to generative AI models as a means of synthesizing additional samples using existing, publicly available datasets. By leveraging the capabilities of generative AI, researchers can generate more training data, which is crucial for improving the performance of predictive AI systems used in the healthcare sector.

In a study by [2], the authors introduced a framework called BrainGAN that generates brain MRI images using VanillaGAN and DCGAN and validates the generated datasets using deep transfer learning models. The framework has four phases: Dataset Collection, Image Generation, Application of deep transfer learning models, and Generated Image Validation. Using this framework, the authors were able to improve the accuracy of automatic tumor detection in MRI scans.

Generative AI models have the capability to not only generate novel medical images but also perform image-to-image translation [26] and enhance images through super-resolution techniques [11]. These models can be trained on large datasets of medical images to learn the underlying patterns and relationships between different types of images. This allows them to translate between different modalities, and enhance low-resolution images to improve their quality.

In a study by [24] the authors used a semi-supervised deep learning approach with a generative adversarial network (GAN) to enhance low-resolution CT images to high-resolution. They used cycle-consistency, joint constraints, and various neural network techniques to facilitate structural preservation and feature extraction. Their model was validated on three large-scale CT datasets and showed promising results compared to other, non-GAN-based state-of-the-art methods.

In summary, generative AI is a rapidly growing field that has captured the interest of both deep learning. It has the potential to solve several issues that affect the performance of predictive AI models, such as lack of data or the inconsistent resolution and quality of medical images. By addressing these challenges, generative AI can improve the accuracy and reliability of predictive AI models, making them more effective in medical applications.

4 What Are the Pros and Cons of Generative AI for Medical Imaging?

In the field of medical imaging, generative AI has both advantages and disadvantages. One of its key benefits is the ability to generate new, realistic training samples without relying on traditional data collection methods from actual patients, which can pose

ethical concerns. This allows for the development of more accurate and reliable predictive models while preserving patient privacy [23].

In addition to generating new training samples, generative AI can also be used to enhance the resolution of low-quality input images. This is particularly useful in situations where images may vary in resolution and quality. By reconstructing higher-resolution samples from low-quality inputs, generative AI can improve the accuracy and reliability of predictive models or enhance the features of certain regions in the input image for clinicians to make decisions [1].

Generative AI has also been used in several instances to carry out image inpainting, which is the process of restoring damaged medical images back to their original version. By leveraging advanced algorithms and techniques, generative AI can fill in missing or damaged parts of an image to create a complete and accurate representation [19].

On the other hand, there are also some challenges associated with using generative AI in medical imaging. One such challenge is the high demand for computational resources during the training of generative AI models. This can be a significant hurdle for organizations with limited resources and may require investment in specialized hardware and infrastructure to support the development and deployment of generative AI models [10].

Another potential disadvantage of using generative AI in medical imaging is that it can be prone to bias and discrimination. This can occur if the models are trained on healthcare data that is not representative of the population they are intended to serve. This can result in inaccurate or unfair predictions for certain groups of patients [21].

In addition, generative AI models can sometimes generate samples with unwanted artifacts or errors. This can occur when the model hallucinates or generates content that is not based on the input data. This can result in inaccurate or misleading outputs, which can be particularly problematic in medical imaging where accuracy and reliability are critical [18]. Table 3 provides a summary of the pros and cons of using generative AI.

Table 3 A summary of the pros and cons of using generative AI in the medical imaging

Pros	Cons
Able to generate new and unique samples	High computational resources are required
Can restore damaged images or images with missing information through inpainting	Generated samples may exhibit bias
Can improve the resolution of input images	Generative AI models may sometimes generate content that is not based on the input data (hallucination)

5 How Generative AI Has Improved Medical Imaging: A Showcase of Successful Applications

5.1 Generating Mammogram Images with Contextual Information Using GANs

Shen et al. [17] proposed a new GAN-based approach to synthesize mammogram images with realistic lesions. Their method consists of two steps: first, they generate multiple mammogram images from a single input image, second, they insert synthetic lesions into the generated images using a contextual infilling technique. They aimed to address the challenge of limited availability of annotated lesion data, which hampers the development and performance of deep learning models for mammogram analysis.

To generate realistic mass images, the authors utilized the DCGAN architecture, which is a type of generative adversarial network (GAN) that can produce high-quality images from random noise. The authors trained the DCGAN to generate both mammogram images and segmentation masks that indicate the location of the lesion in a given mammogram image. The segmentation masks are binary images that have the same size as the mammogram images, where white pixels represent the lesion region and black pixels represent the background. The authors used the segmentation masks as a condition for the DCGAN to ensure that the generated mass images have accurate mask annotations. Thus, the synthesized images can be applied in downstream detection tasks.

To evaluate their method, the authors used the generated dataset to train a U-NET segmentation network, which is a type of deep neural network that can segment regions of interest in images. The U-NET network was trained to detect potential lesions in mammogram images by comparing the input images with the corresponding segmentation masks. The authors reported that the U-NET network trained on the synthesized dataset achieved better performance than the same network trained on a public dataset with traditional augmentation methods, such as rotation, flipping and scaling.

According to their results, their method improved the detection rate by 5.03% and reduced the false positive rate by 2.12% compared to using only real data [17].

5.2 Augmenting Medical Images Using Semi-supervised GANs and Attention Mechanism

Another interesting example of using GANs to solve the data disparity problem is a study by [14]. The authors utilized CycleGAN, a method proposed by [26], to generate MRI samples with and without lesions. Their goal was to create a larger dataset of both positive and negative MRI images for tumor classification.

CycleGAN is a type of generative adversarial network (GAN) that can learn to translate images from one domain to another without paired data. GANs are a powerful tool for medical image synthesis and can enhance the performance of deep learning models for medical image analysis. Qi et al. [14] applied CycleGAN to MRI samples without lesions to synthesize lesions in them, and to MRI samples with lesions to remove lesions from them.

The authors used their method to create more MRI samples from three small, public MRI datasets: BrainWeb, BRATS 2015 and ISLES 2015. These datasets contain MRI images of brain tumors with different modalities and annotations. They trained a ResNet18 classifier with the real and the new samples to classify them as positive or negative. They also trained the same classifier with only the real samples that had some basic augmentation, such as rotation, flipping, and translation.

The authors compared their method with using only real data and found that their method improved the accuracy of tumor classification. They reported that their method increased the accuracy by 2.8%, 3.6% and 4.1% on BrainWeb, BRATS 2015 and ISLES 2015 datasets respectively.

5.3 EndoVAE: An Innovative Variational Autoencoder for Endoscopic Image Generation

The authors of EndoVAE [5] proposed a variational autoencoder model that can generate realistic and diverse endoscopy images, which can be used to augment an existing dataset or even replace it completely. The main goal of this study is to create a large and varied dataset of synthetic endoscopy images that can be used for training or testing other models without violating any privacy or ethical regulations.

The authors chose to use VAEs instead of GANs because GANs have several well-known issues that make them difficult to train and prone to produce low-quality or repetitive images, especially in medical imaging domains where the data availability is limited.

Instead of training an additional model to detect abnormalities in endoscopy images, the authors evaluated their model for diversity and quality using metrics like Frechet Inception Distance (FID), Structural Similarity Index (SSIM) and Image Patch Similarity (LPIPS) and showed that their model outperformed other generative models such as WGAN-GP and StyleGAN2.

5.4 Restoring Medical Images with Variational Autoencoders

Generative AI models have many potential applications in different domains. One of them is retrieving medical images that are relevant and similar to a given query image, which can help doctors and researchers find similar cases or studies.

A recent study by [3] demonstrated that variational autoencoders, a type of generative model that learns a latent representation of the data, can effectively retrieve medical images based on their visual similarity. They used an unsupervised approach based on variational autoencoders (VAEs) to learn deep features for content-based medical image retrieval (CBMIR).

They showed that VAEs can retrieve images that are more visually similar than the predominant supervised approach based on classification, and can also be used in combination with the classification approach to achieve better results. They also proposed a new evaluation method based on hidden classes that reflects the visual similarity of the retrieved images better than the traditional evaluation based on labels. They tested their approach on two medical image datasets and compared it with other methods, such as PCA, LDA, and CNNs.

5.5 Brain Imaging Made Easy with Latent Diffusion Models

Compared to GANs and VAEs, diffusion models can produce more realistic images which is part of the reason, they are becoming more popular among computer vision researchers.

One such application is brain imaging generation, which was explored by [13] in their paper titled Brain Imaging Generation with Latent Diffusion Models. They trained a diffusion model on high-resolution 3D brain images from the UK Biobank dataset and used covariates such as age and sex to condition the model on the probabilistic distribution of brain images.

They also showed how to manipulate the diversity of the generated data by changing the conditioning variables. They released a dataset of 100,000 synthetic brain MRIs for further scientific use.

The authors followed a series of steps to preprocess and train their latent diffusion models on the UK Biobank dataset of T1w MRI images. First, they cropped, resampled, and normalized the images to a common space and intensity range. Then, they divided the dataset into training (90%) and validation (10%) sets and used a batch size of 16 images for training. Next, they designed their models using a U-Net architecture for both the encoder and decoder networks, and a fully connected network for the predictor network.

5.6 Creating Realistic 3D Medical Images with Denoising Diffusion Models

In their paper titled “Denoising diffusion probabilistic models for 3D medical image generation”, [9] introduced a novel method for synthesizing realistic MRI and CT images from textual input or noise using denoising diffusion probabilistic models.

They evaluated their method on the task of breast segmentation, which is a challenging problem in medical image analysis. They compared their method with other state-of-the-art generative models, such as GANs and VAEs, and showed that their method can significantly improve the performance of breast segmentation models by increasing the dice score, which is a common metric for measuring the overlap between the predicted and ground truth segmentations, from 0.91 to 0.95, which is a large improvement of 0.4.

They also assessed the quality of the generated images by asking two radiologists to rate them on three criteria: realistic image appearance, anatomical correctness, and consistency between slices. They found that their method can produce high-quality images that are indistinguishable from real images by human experts. Furthermore, they demonstrated that the generated images can be used for self-supervised pre-training and enhance the performance of breast segmentation models when data is scarce, which is a common scenario in medical imaging research.

6 Generative AI Outlook: A Game-Changer for Healthcare and Biomedical Research

6.1 Key Takeaways and Findings of the Chapter

In this chapter, we have discussed the state-of-the-art generative AI models and techniques for medical imaging and looked into their current and potential applications, challenges, and opportunities, in relation to medical imaging. The key takeaway and findings of this chapter listed below:

- Generative AI models, such as GANs, VAEs, and diffusion models, can generate realistic and diverse images for medical imaging applications.
- Generative AI models can help overcome the challenges of acquiring and processing large and high-quality medical imaging datasets, such as privacy, ethical, and cost issues.
- Generative AI models can also create novel and synthetic images for diagnosis and treatment purposes, such as image synthesis, augmentation, reconstruction, segmentation, registration, anomaly detection, and classification.
- Generative AI models require careful evaluation and validation to ensure the quality and validity of the generated images, as well as the privacy and security of the patient data.
- Generative AI models also raise ethical and social questions about the use and impact of synthetic data on healthcare and society.
- Generative AI models could sometimes hallucinate; therefore, creating unrealistic or corrupted samples.

6.2 Recommendations for Generative AI Users and Stakeholders

Generative AI models are constantly developing and improving with new methods and approaches emerging frequently. There are several suggestions for AI users and stakeholders that can be condensed into the following points:

- The medical industry should establish clear guidelines for the use of generative AI, especially on medical imaging.
- The data collection and processing of patient data should be transparent and informed consent should be obtained for how the data will be used.
- User-friendly and accessible platforms for synthesizing medical images should be developed for clinicians, researchers, and other stakeholders.
- The development and evaluation of generative AI models for medical imaging should follow rigorous and transparent standards and protocols to ensure the trustworthiness and reliability of the models.
- The users and stakeholders of generative AI for medical imaging should be aware of the potential risks and benefits of using synthetic data and engage in ethical and social dialogue and education to foster responsible and informed decision-making.

6.3 Future Directions for Generative AI Research for Medical Imaging

Generative AI is a flourishing and lively research area that has the potential to revolutionize the healthcare industry. By creating new data or content from existing data or content, generative AI can help improve the productivity and excellence of the work of healthcare professionals and deep learning researchers who are interested in this field. For example, generative AI can create synthetic images that are close to real images, improve medical imaging quality, reduce administrative burden, interpret unstructured data, enable medical robots, assist in disease diagnosis and screening, and facilitate drug discovery and development.

As artificial intelligence progresses and more data becomes available, the quality of generative AI models will increase over time and new, advanced models that can create completely original samples will emerge. Moreover, these models will be able to generate content that is consistent with the context, style, and preferences of the stakeholders.

In addition, we expect that generative AI models will become more available to clinicians and researchers to produce as many synthetic samples as they need. These samples can then be used to train predictive AI models or for other purposes that are not related to AI.

For instance, synthetic samples can be used to augment real data, enhance privacy protection, evaluate model robustness, or conduct experiments. By using generative

AI models, clinicians and researchers can overcome the limitations of real data and explore new possibilities in healthcare.

References

1. Alamir, M., & Alghamdi, M. (2022). The role of generative adversarial network in medical image analysis: An in-depth survey. *ACM Computing Surveys*, 55(5). <https://doi.org/10.1145/3527849>.
2. Alrashedy, H. H. N., Almansour, A. F., Ibrahim, D. M., & Hammoudeh, M. A. A. (2022). BrainGAN: Brain MRI image generation and classification framework using GAN architectures and CNN models. *Sensors*, 22(11). <https://doi.org/10.3390/s22114297>.
3. Alves, C., & Traina, A. J. M. (2022). Variational autoencoders for medical image retrieval. *16th International Conference on INnovations in Intelligent SysTems and Applications, INISTA 2022*. <https://doi.org/10.1109/INISTA55318.2022.9894251>.
4. Çallı, E., Sogancioglu, E., van Ginneken, B., van Leeuwen, K. G., & Murphy, K. (2021). Deep learning for chest X-ray analysis: A survey. *Medical Image Analysis*, 72, 102125. <https://doi.org/10.1016/j.media.2021.102125>.
5. Diamantis, D. E., Gatoula, P., & Iakovidis, D. K. (2022). EndoVAE: Generating endoscopic images with a variational autoencoder. *IVMSP 2022–2022 IEEE 14th Image, Video, and Multi-dimensional Signal Processing Workshop*, 1–5. <https://doi.org/10.1109/IVMSP54334.2022.9816329>.
6. Gassenmaier, S., Küstner, T., Nickel, D., Herrmann, J., Hoffmann, R., Almansour, H., Afat, S., Nikolaou, K., & Othman, A. E. (2021). Deep learning applications in magnetic resonance imaging: Has the future become present? *Diagnostics*, 11(12), 1–11. <https://doi.org/10.3390/diagnostics11122181>
7. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139–144. <https://doi.org/10.1145/3422622>
8. Karras, T., Laine, S., & Aila, T. (2021). A style-based generator architecture for generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12), 4217–4228. <https://doi.org/10.1109/TPAMI.2020.2970919>
9. Khader, F., Mueller-Franzes, G., Arasteh, S. T., Han, T., Haarburger, C., Schulze-Hagen, M., Schad, P., Engelhardt, S., Baessler, B., Foersch, S., Stegmaier, J., Kuhl, C., Nebelung, S., Kather, J. N., & Truhn, D. (2022). Medical diffusion: Denoising diffusion probabilistic models for 3d medical image generation. *Scientific Reports*, 1–12. <https://doi.org/10.1038/s41598-023-34341-2>.
10. Khayatkhoei, M., & Elgammal, A. (2022). Spatial Frequency bias in convolutional generative adversarial networks. *Proceedings of the 36th AAAI Conference on Artificial Intelligence, AAAI 2022*, 36, 7152–7159. <https://doi.org/10.1609/aaai.v36i7.20675>.
11. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., & Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. *Proceedings—30th IEEE conference on computer vision and pattern recognition, CVPR 2017, 2017-Janua*, pp. 105–114. <https://doi.org/10.1109/CVPR.2017.19>
12. Mirza, M., & Osindero, S. (2014). *Conditional generative adversarial nets*, pp. 1–7. <http://arxiv.org/abs/1411.1784>.
13. Pinaya, W. H. L., Tudosi, P. D., Dafflon, J., Da Costa, P. F., Fernandez, V., Nachev, P., Ourselin, S., & Cardoso, M. J. (2022). Brain imaging generation with latent diffusion models. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*, 13609 LNCS, pp. 117–126. https://doi.org/10.1007/978-3-031-18576-2_12.

14. Qi, C., Chen, J., Xu, G., Xu, Z., Lukasiewicz, T., & Liu, Y. (2020). *SAG-GAN: Semi-supervised attention-guided gans for data augmentation on medical images*. <http://arxiv.org/abs/2011.07534>.
15. Radford, A., Metz, L., & Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th International conference on learning representations, ICLR 2016—conference track proceedings*, pp. 1–16.
16. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. (2016). Generative adversarial text to image synthesis. In *33rd International Conference on Machine Learning, ICML 2016*, 3, 1681–1690.
17. Shen, T., Hao, K., Gou, C., & Wang, F. (2021). Mass image synthesis in mammogram with contextual information based on GANs. *Computer Methods and Programs in Biomedicine*, 202, 106019. <https://doi.org/10.1016/j.cmpb.2021.106019>
18. Shin, Y., Yang, J., & Lee, Y. H. (2021). Deep generative adversarial networks: Applications in musculoskeletal imaging. *Radiology: Artificial Intelligence*, 3(3). <https://doi.org/10.1148/ryai.2021200157>.
19. Siavelis, P. R., Lamprinou, N., & Psarakis, E. Z. (2020). An improved GAN semantic image inpainting. In *Lecture notes in computer science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 12002 LNCS*. Springer International Publishing. https://doi.org/10.1007/978-3-030-40605-9_38.
20. Sun, J., Liao, Q. V., Muller, M., Agarwal, M., Houde, S., Talamadupula, K., & Weisz, J. D. (2022). Investigating explainability of generative AI for code through scenario-based design. *International Conference on Intelligent User Interfaces, Proceedings IUI*, pp. 212–228. <https://doi.org/10.1145/3490099.3511119>.
21. Tan, S., Shen, Y., & Zhou, B. (2020). *Improving the fairness of deep generative models without retraining*. <http://arxiv.org/abs/2012.04842>.
22. Tanwar, S., Vijayalakshmi, S., Sabharwal, M., Kaur, M., Alzubi, A. A., & Lee, H. N. (2022). Detection and classification of colorectal polyp using deep learning. *BioMed Research International*, 2022. <https://doi.org/10.1155/2022/2805607>.
23. Tom, E., Keane, P. A., Blazes, M., Pasquale, L. R., Chiang, M. F., Lee, A. Y., & Lee, C. S. (2020). Protecting data privacy in the age of ai-enabled ophthalmology. *Translational Vision Science and Technology*, 9(2), 1–7. <https://doi.org/10.1167/tvst.9.2.36>
24. You, C., Cong, W., Vannier, M. W., Saha, P. K., Hoffman, E. A., Wang, G., Li, G., Zhang, Y., Zhang, X., Shan, H., Li, M., Ju, S., Zhao, Z., & Zhang, Z. (2020). CT super-resolution GAN constrained by the identical, residual, and cycle learning ensemble (GAN-CIRCLE). *IEEE Transactions on Medical Imaging*, 39(1), 188–203. <https://doi.org/10.1109/TMI.2019.2922960>
25. Zhao, W., Jiang, W., & Qiu, X. (2021). Deep learning for COVID-19 detection based on CT images. *Scientific Reports*, 11(1), 1–12. <https://doi.org/10.1038/s41598-021-93832-2>
26. Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision, 2017-Octob*, pp. 2242–2251. <https://doi.org/10.1109/ICCV.2017.244>.
27. Zohny, H., McMillan, J., & King, M. (2023). Ethics of generative AI. *Journal of Medical Ethics*, 49(2), 79–80. <https://doi.org/10.1136/jme-2023-108909>

Generative AI in Medical Imaging and Its Application in Low Dose Computed Tomography (CT) Image Denoising



Luella Marcos, Paul Babyn, and Javad Alirezaie

Abstract Deep learning techniques have made its way to the medical field. Medical images are essential tools for visualizing internal body structures up to cellular levels. X-ray computed tomography (CT) is a widely used non-invasive medical modality for patient diagnosis. Harmful effects of cumulative amounts of radiation exposure to patients undergoing CT scan have been recorded which includes hair loss, cancer and other illnesses. The “As Low as Reasonably Achievable” (ALARA) principle was developed with the purpose of minimizing the radiation dose to patients. This chapter discusses the implementation of artificial intelligence to devices for the reconstruction of CT images affected by the reduction of the radiation. The corrupted CT images have noticeable noise and artifacts that causes inaccuracies of medical diagnosis. One of the robust deep learning models for LDCT restoration is the Generative Adversarial Networks (GAN). This study shows a simple GAN architecture that aims to minimize edge over-smoothing, image texture enhancement and preservation of structural details of the medical images. Further, a benchmark testing was done to show the performance of the network compared with other state of the art models (SOTA). In addition, ablation experiments for the modules used in the network and loss functions used for the training procedure are also presented.

Keywords Medical imaging · Low dose computed tomography · Image denoising · GAN · ResNets · Deep learning · Ionizing radiation

L. Marcos · J. Alirezaie (✉)

Toronto Metropolitan University (formerly Ryerson University), 350 Victoria Street,
Toronto, ON M5B 2K3, Canada
e-mail: javad@torontomu.ca

L. Marcos

e-mail: lgmarcos@torontomu.ca

P. Babyn

University of Saskatchewan, Region University Hospital, 103 Hospital Drive, Saskatoon, SK S7N
0W8, Canada
e-mail: paul.babyn@saskatoonhealthregion.ca

1 Introduction

In medical image processing, enhancing the quality of the images is a challenging task where high-resolution images are being generated from low-resolution images. One example of this task can be seen in X-ray computed tomography (CT) image denoising. Computed tomography (CT) images are widely known for medical diagnosis and exposure of patients to radiation is required to obtain the images.

The 1945 atomic bombing in Hiroshima and Nagasaki, Japan had started the studies regarding the risks caused by ionizing radiation to humans [1]. During this incident, investigations about the health of the 25,000 survivors had been done and findings show that victims, exposed to less than 50 millisievert(mSv) of radiatioin, had increased chance of having cancer [1]. This is comparatively close to the accumulated radiation received by patients undergoing three or more CT scans [2].

In order to minimize the risk of radiation exposure of patients undergoing CT scans for medical diagnosis, researchers recommend the use of low-dose CT (LDCT) images then applying image denoising methods to handle the noise produced from low radiation. Deep learning techniques have been developed for this problem, in which artificial neural networks are being utilized. Deep learning methods have been providing more reliable outcomes as they evolve through the years. Taking advantage of the capability of graphics processing unit (GPU) to perform parallel computing has accelerated the training process especially when the network depth increased.

As promising as it sounds, a well known limitation for deep learning methods is the availability of the medical images readily available for study. In order for deep learning methods learn how to perform a specified task, a huge amount of data is needed. Researchers have been finding ways to address this issue and one solution is using unsupervised learning procedures. Generative Adversarial Network (GAN) is the most conventional unsupervised learning algorithm and takes advantage of using supervised loss for its training. Therefore, this chapter focuses on implementing this network for CT denoising.

To understand this topic better, detailed explanation of computed tomography, image denoising techniques and basic deep learning concepts are presented. Further, results and analysis of denoised CT images from an implemented simple GAN residual network are also included in this chapter to have a complete idea.

2 Computed Tomography

Computed Tomography were first developed by Sir Godfrey Newbold Hounsfield in 1972 [3]. CT scanners have come a long way with the fast advancement of new technologies. Nowadays, CT scans can now take images in less than one second which is much faster compared to the first generation CT which can take image slice within 300 s. That being said, the decline of the CT imaging usage was expected with the introduction of Magnetic Resonance Imaging (MRI). However, it continued to be

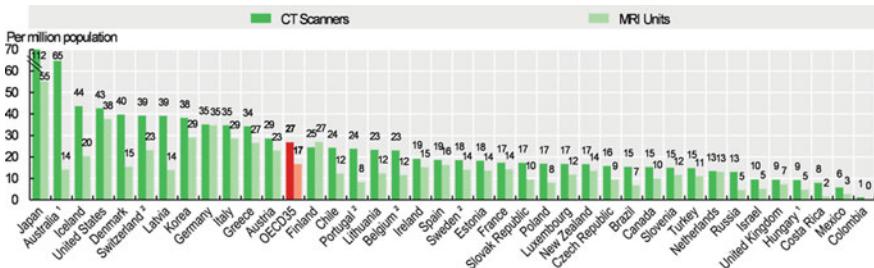


Fig. 1 Availability and usage of CT scanners and MRI units to (OECD) countries. Photo courtesy of OECD [5]

the most commonly used image diagnostic tool for it is the fastest and cheapest whilst being capable of producing accurate diagnostic images. According to the Canadian Agency for Drugs and Technologies in Health (CADTH), there are currently 378 MRI units and 549 CT units that are being used in Canada as of 2022 [4]. Not only in Canada, but also in other parts of the world that mostly used CT scanners more than MRI units as of 2017 as shown in the graph in Fig. 1 [5]. That being said, CT imaging has continued to develop and new variation such as positron and single-photon emission CT (PET/CT), which now relates to molecular imaging.

2.1 How Do CT Devices Work?

Commercial CT machines generate the images of human internal structure with the help of X-ray beams through a collimator. This process is illustrated in Fig. 2. A patient is being delivered through the machine by the motorized platform in Fig. 2a, then as the patient passes through the gantry (Fig. 2b), X-ray detectors and beams rotates around the body to capture the regions of interests as demonstrated in Fig. 2c. Photodiodes in this process help convert the obtained photons into electric signals and transmitted to a computer. From this, CT scans can produce 2D or 3D cross-sectional images via computer processing. The Hounsfield Unit (HU), unit for characterizing radiation attenuation in tissues, is dependent on the intensity of these photons [6]. This Hounsfield scale is used for comparing tissues being observed to those that are considered to be healthy tissues [7] as summarized in Table 1. On the CT image in spatial domain, areas that contain high calcium (i.e. bones and teeth) will appear white on the image, and associated with high radiation attenuation. In contrast, areas with low radiation attenuation or weak absorption of radiation would appear black [8].

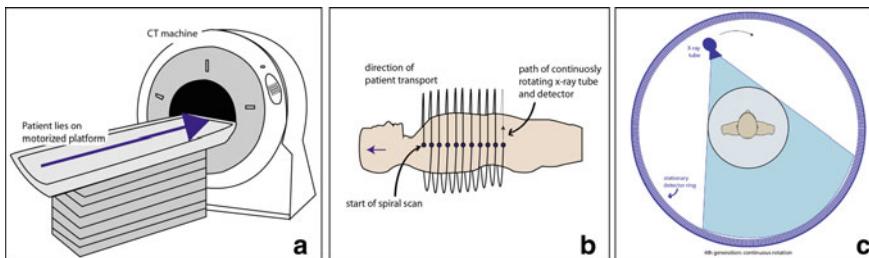


Fig. 2 Computed Tomography (CT) scanning procedure: **a** Modern CT Machine; **b** X-ray direction once patient is being transported into the machine and; **c** Top-view of the 4th generation fan beam CT scanner. Photo courtesy of Marcos [6]

Table 1 Hounsfield units of different tissues and materials [9]

Tissue	Hounsfield unit (HU)
Air	-1000
Lung	-700 to -600
Fat	-120 to -90
Water	0
Abscess/pus	0 or +20, +40 or +45
Blood	+30 to +45
Muscle	+35 to +55
Bone	+700 to +3000

2.2 X-Ray Radiation and Risks

As mentioned previously, acquiring images of the internal human structures require X-ray radiation. X-ray radiation is produced from the sudden decline in speed of fast-moving electrons as they collide with an anode. Most of the energy released from this collision is released as heat while few is converted to X-rays [10]. X-ray radiation wavelength ranges approximately $\times 10^{-10}$ m in the electromagnetic spectrum, between ultraviolet (UV) and Gamma rays [1]. For CT scans, 0.018 to 0.025 nm wavelength is needed for obtaining the images with a mean acceleration energy of 50–70 Kilo electron volts (KeV) [11].

The exceptional images obtained from the CT machines make them valuable for medical diagnosis. However, the potential overuse of medical imaging was recognized by the American Association of Physicists in Medicine (AAPM) [12]. During CT scanning procedure, small amounts of the ionizing radiation are being absorbed into the bodies of patients. Some may argue that radiation from this procedure only contributes about 4% of the patient exposure, but the accumulated dose from undergoing the procedure multiple times can reach up to 35% [1]. According to the Archives of International Medicine [12, 13], as the CT examination increases, cases of radiation-induced cancer also arises. Hence, this study falls under the ALARA principle which stands for “As low as (is) reasonably achievable” that aims to maintain

or reduce human exposures to ionizing radiation as introduced by the International Commission on Radiological Protection (ICRP) [14].

3 CT Image Denoising

Noise is a type of image corruption that deteriorates the quality of medical images and affects the diagnosis of medical professionals to patients [15]. Noise usually include blurriness, information loss, formation of new artifacts or unclear textural patterns on the medical images. Such corruptions can be caused by the limitations of hardware or software components where the images are being acquired. That being said, the effects of different noises depend on the causes and the effects may vary. For example, random noise caused by the detection of X-ray quanta with finite amount in the project can appear as oscillations in the image [16]. Moreover, the most common noise that can be seen in CT images is quantum noise, which is mainly due to the lack of X-ray photons and as a result, edge-smoothing and structural detail loss with low contrast is visible in the spatial domain of the CT image. This noise can be modelled as Poisson distribution [17]. Other factors that can affect visual quality of CT images are slice thickness, patient size and the type of algorithm used for reconstruction that could affect the distribution intensities of image pixels.

As a continuation of the discussion on how CT devices work in Sect. 2.1, this section also shows how CT image slices are being obtained. Shown in Fig. 3, CT imaging can be analyzed in either the projection domain or in the image domain [15]. This will serve as a basis for CT image denoising.

In this context, image denoising can be interchangeably called CT image reconstruction, which can be classified into two categories. On one hand, pre-reconstruction CT denoising, uses projection data for statistical analysis of the noise characteristics present in the sinogram domain of the images then removing them. Despite being effective, noise artifacts are still visible and computational cost is high. Further, projection data is not easily available from the commercial CT scanners. On the other hand, the CT post-reconstruction involves low-dose CT images used as inputs in the

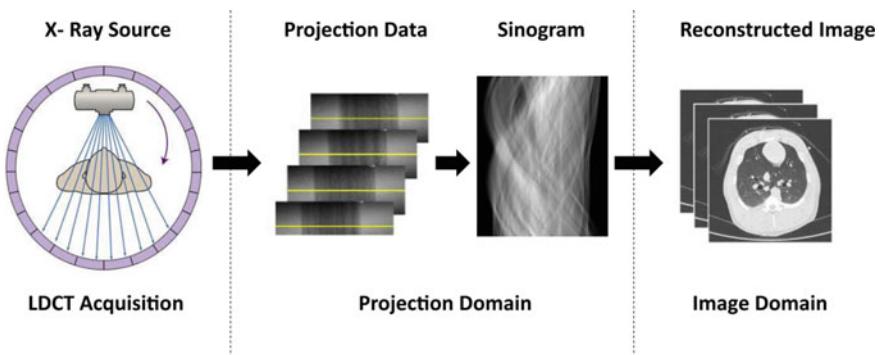


Fig. 3 CT imaging domains for analysis. Photo courtesy of Kulthilake et al. [15]

system to produce high quality images that are close to those normal-dose CT images (NDCT). This process takes advantage of deep learning techniques.

Various adaptive medical image reconstruction exist but the main focus of this chapter is AI-based models, which includes shallow learning: artificial neural networks (ANN) and functional link ANN (FLANN) [18]; and lastly, deep learning: convolutional neural network (CNN) [19], residual Networks (ResNet) [20, 21], Auto-encoder [22], and Generative Adversarial Network (GAN) [23]. For deep learning, combination of these networks are common to improve the denoising network of the image processing system.

4 CT Denoising AI Networks

State-of-the-art (SOTA) image denoising models using deep learning techniques are booming in the medical imaging field due to their fast and reliable outcomes. For this chapter, a simple GAN architecture formed using residual and auto-encoder based network is investigated to show the growing importance of machine learning in this research field.

A generative adversarial network (GAN)-based denoising framework usually consists of a generator, feature extractor and a discriminator as illustrated in Fig. 4. During phase 1 or training procedure, a random pool from the compiled output fake denoised images from the generator, ground truth and LDCT images will be fed through the discriminator. Then, a score on how the generated fake images is similar to the ground-truth is provided from the discriminator. The network parameters in the generator will be adjusted according to the discriminator score and the loss values obtained from the feature extractor component. Once the training is done, the generator network is used as the main unsupervised denoising network for phase 2 or the testing process. These three GAN modules can be structured differently based

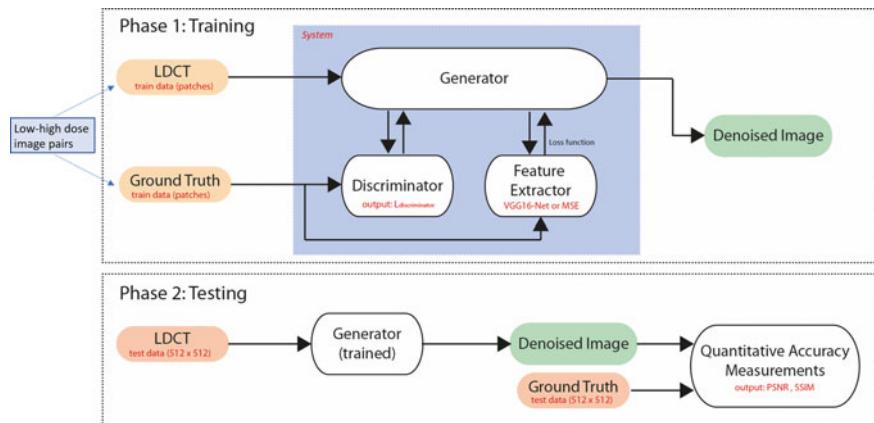


Fig. 4 Schematic diagram of a denoising system using GAN architecture

on the mentioned deep learning-based models. Thus, GAN-based model is an ideal technique to use for this particular problem due to this flexibility.

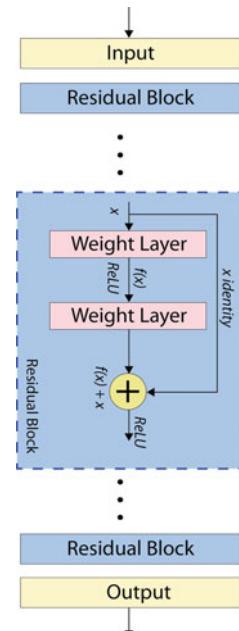
It is important to note that the main purpose of this architecture is to solve a min-max optimization where the generator, G , tries to minimize the objective function while the discriminator, D maximizes it. Mathematically, the parameters of G map the samples (z) from the noise distribution $p(z)$, and D shows the probability that sample (x) belongs to true data $p_{data}(x)$. This relationship can be represented as follows:

$$\min_G \max_D GAN(D, G) = E_x p_{data}(x) [\log D(x)] + E_z p_z(z) [\log (1 - D(G(z)))] \quad (1)$$

By simply looking at this relationship, it can be deduced that as the accuracy of the discriminator increases, the performance of the generator gets worse during the training process, causing vanishing gradient problem. In other words, making the whole system unstable so balancing the performance of the two components is the crucial part of implementing this network. Although GAN framework can certainly preserve structural information of images, problems like blurring is noticeable and its complex structure is also prone to system instability due to its oscillating number of parameters during the training process. Hence, for the simple experiment for this chapter, the concept of residual learning network is adapted.

Residual learning—based networks (ResNets) are simpler and more stable. The main key for this networks is the use of skip connection between the pre- and post convolutional layers during the denoising process. Figure 5 shows the core blocks of implementing a residual network that demonstrates this concept. The input is

Fig. 5 Core blocks of a ResNet architecture



fed-forward and added to the output of two consecutive network layers, $f(x)$. This helps with the preservation of the structural details of the image being denoised. A typical residual encoder-decoder CNN (RED-CNN) follows this idea. Along with the skip-connections, it also uses auto-encoder which compresses the input data and encodes it, then learns to reconstruct the data by mapping the encoded representation [15]. Various models based on this have been made such as parameter-dependent framework(PDF)-based RED-CNN for modulating feature maps, U-Net merged with ResNet for learning both local and global features, and a dilated residual learning with and edge detection layer (DRL-E-MP) for capturing fine details at the image boundaries [21].

Therefore, the denoising network used for this experiment applied the combined concept of GAN and RED-CNN which is referred to as RED-GAN for simplicity. The concept of RED-CNN for its generator, Fig. 6a, while a simple ResNet is applied on the discriminator as shown in Fig. 6b. All layers uses convolution operation with kernel size of 3×3 with strides = 2 and **n64** filters, followed by batch normalization (BN) and rectified linear unit (ReLU) layers. Having said that, the last layers have a kernel size of 1×1 and uses only 1 filter and strides = 1, and does not follow by any BN or ReLU operations.

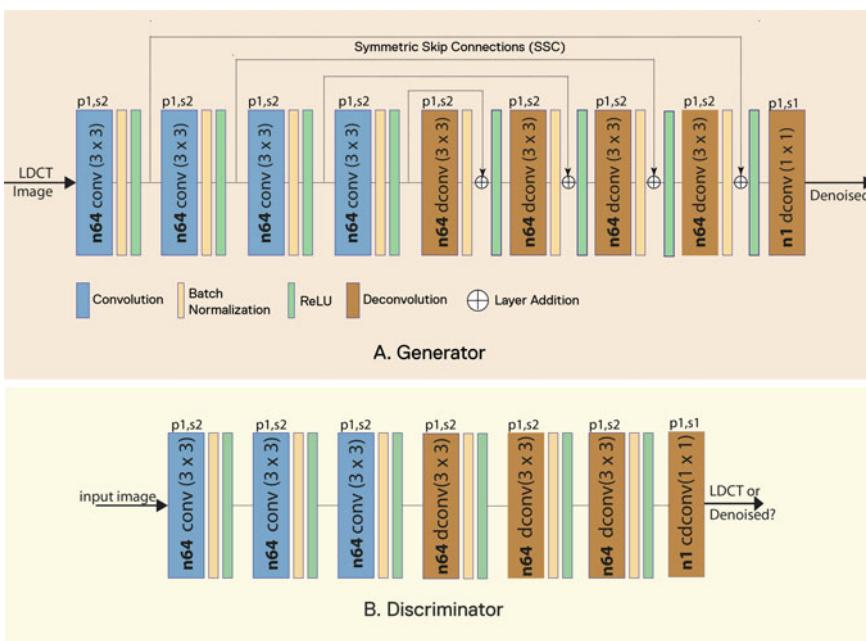


Fig. 6 RED-GAN architecture used for this study

5 Denoising Metrics

Noise in natural images can often be suppressed due to its type-dependency. In contrast, the data distribution of low-dose CT is a combination of electronic Gaussian and quantum Poisson noises that change significantly over the CT image. As a result, typical denoising approaches are unsatisfactory for LDCT images. However, because deep learning-based approaches are insensitive to the statistical distribution of noise, the low dose CT problem can be solved based on the following model. Assume that $x \in R^{m \times n}$ is a low-dose CT image and $y \in R^{m \times n}$ is the normal-dose image

$$x = \varphi(y) \quad (2)$$

where $\varphi : R^{m \times n} \rightarrow R^{m \times n}$ denotes the corruption in the image caused by quantum noise. Thus, the noise suppression process is simplified to the effort of determining a function f such that

$$f = \|f(x) - y\|_2^2 + \Omega(\theta) \quad (3)$$

where f is regarded to be the best approximation of φ^{-1} and $\Omega(\theta)$ is regularization term in which θ is regarded as the parameters of f . One important accuracy metric that are commonly used for image denoising is the signal-to-noise ratio (SNR), which is defined as the ratio between the power and the noise. Mathematically, this can be represented as:

$$\begin{aligned} \text{SNR} &= \frac{P_{\text{signal}}}{P_{\text{noise}}} = \left(\frac{A_{\text{signal}}}{A_{\text{noise}}} \right)^2 \\ \text{SNR(dB)} &= 10 \log \frac{P_{\text{signal}}}{P_{\text{noise}}} = 20 \log \left(\frac{A_{\text{signal}}}{A_{\text{noise}}} \right) \end{aligned} \quad (4)$$

where the power of the noise and signal are P_{signal} and P_{noise} and their amplitudes are A_{signal} and A_{noise} respectively.

This objective function can be formulated and used through different distance metrics. Here, simply the mean square error is displayed. However, there are also other metrics that can be employed for this purpose, which will be discussed in the following sections.

5.1 Per Pixel Loss Function

Mean-squared error (MSE) or “per-pixel loss” is one of the most common objective functions for it ensures differentiability, symmetry and convexity between LDCT and ground-truth image pairs, $(x_j, y_j)_{j=1}^N$, at a pixel level by minimizing the following L_{MSE} function with respect to the model parameters (θ) [15].

$$L_{MSE}(\theta) = \frac{1}{N} \sum_{i=1}^N \|f(x_i; \theta) - y_i\|_F^2. \quad (5)$$

5.2 Evaluation Metrics

The results of CT denoising methods can generally be evaluated in three ways: reader study (visual effect evaluation), quantitative measurements, and textural measurements. Radiologists, using the reader study technique, independently examine the visual effects of the reconstructed images by considering several factors such as whether the model can preserve structural details and suppress the noise. The peak signal-to-noise ratio (PSNR), and the structural similarity index (SSIM) are all quantitative metrics that will be used to test the accuracy of the output images from the ground-truth.

Peak-Signal-to-Noise Ratio. Firstly, PSNR is the metric used to evaluate the algorithms' noise suppression ability, which is based on SNR explained previously. Mathematically,

$$PSNR(x, y) = 10 \log_{10} \frac{(\max(x_j))^2}{MSE} \quad (6)$$

Structural Similarity Index. Next is the structural similarity index (SSIM) which compares images' structural information, such as texture, contrast and luminance. Further, to better understand the deep-learning outcomes for noise reduction in low-dose CT, textural measurements are obtained by calculating the reconstructed image's statistical properties such as the mean CT number (Hounsfield Unit), standard deviation, uniformity, and entropy. Denoising approaches should ideally yield generated CT images with statistical attributes near the equivalent NDCT image as possible. This can be represented as:

$$SSIM(z, z^*) = \frac{(2\bar{z}\bar{z}^* + c_1)(2\sigma_{zz^*} + c_2)}{(\bar{z}^2 + \bar{z}^{*2} + c_1)(\sigma_z^2 + \sigma_{\bar{z}}^2 + c_2)} \quad (7)$$

where \bar{z} and \bar{z}^* represent the mean values of z and z^* , σ_z and $\sigma_{\bar{z}}$ represent the variances of z and z^* , σ_{zz^*} denotes the covariance of z and z^* , and c_1 and c_2 represent the constants.

5.3 Dissimilarity Index Loss Function

Since the objective is to maximize this value close to 1, using SSIM as a loss function would give higher losses so therefore, Structural Dissimilarity can be used which is the SSIM equivalent of a kernel function shown in Eq. 8.

$$L_{DSSIM}(\theta) = \frac{1}{N} \sum_{i=1}^N (1 - SSIM(f(x_i; \theta) - y_i))/2 \quad (8)$$

where $\hat{y}(\theta) = f(x_i; \theta)$ is the denoised image and y is the corresponding ground truth image. Theoretically, this helps with the preservation of the textural details of the image.

5.4 Overall Loss Function

The loss functions mentioned play a big role in optimizing the CT denoising models and largely affects the quality of denoised images. An ablation experiment can be done to investigate the effect of each loss function in the network. In order to complement the characteristics that each loss function offers, the overall objective function that will be used for this experiment can be set as:

$$L(\theta) = \gamma_1 L_{MSE}(\theta) + \gamma_2 L_{DSSIM}(\theta) \quad (9)$$

where γ_1 , and γ_2 are the sum-to-one weights for the two loss components. Each of the weights is determined during the training process, where the maximum value of the losses after each epoch is used for updating the values of the weights. The loss function that obtained the greatest loss would receive a higher scale than the other functions.

6 Experimental Setup

A total of 146 512×512 LDCT-NDCT (normal dose CT) image pairs are used for the CT denoising models. This data comes from the CT simulation of a deceased piglet provided by Yi and Babyn in [24]. The 80% training, 10% testing and 10% validation partition of the data and 64×64 overlapping patches on the training data are also applied. All the data in this work are in 2D slices and all the models are trained for 100 epochs with a batch size of 16 using the Adam optimizer with a learning rate of 0.0002, decay steps of 0.5 and decay rate of 0.9. Further, the experiment is carried out using Python programming in the environment of Tensorflow-Keras API on Windows operating system with NVIDIA GeForce GTX 1080 graphics card.

7 Results and Analysis

The results of CT denoising methods can generally be evaluated in two ways: reader study (visual effect evaluation), and quantitative measurements. Radiologists, using the reader study technique, independently examine the visual effects of the reconstructed images by considering several factors such as whether the model can preserve structural details and suppress the noise. The peak signal-to-noise ratio (PSNR), and the structural similarity index (SSIM) are all quantitative metrics that are used to test the accuracy of the output images from the ground-truth. For this investigation, the validity of using the overall combination of the loss functions is further strengthen by implementing three modifications of the RED-GAN model namely: (i) RED-GAN with only using MSE, (ii) RED-GAN with only DSSIM and finally, (iii) RED-GAN with hybrid loss function. Moreover, benchmark testing is also done by comparing the quantitative and visual results of the experimental model and its variations with a vanilla RED-CNN, self-attention based RED-CNN (RED-SA) [25], and DRL-E.

Figure 7 summarizes the both visual and quantitative results of the experiment using one Piglet CT image slice. Please note that specific regions, marked with green and yellow boxes, on the selected image slice correspond to the structural detail where the differences between the algorithms are the most prominent. Starting with the visual results, it is noticeable that all the image outputs of CT denoising models with only MSE as loss function (Fig. 7c, e, g, i) demonstrated over smoothing along the edges and blurring effects. Based from these results, it can be deduced that MSE fails to capture textural details fully. Adding DSSIM as loss function slightly improved the texture and enhanced the contrast of the results of all the models. This can be supported by looking at the PSNR/SSIM values of the models with MSE+DSSIM loss function displayed in Fig. 7d, f, h, j. Hence, this shows that using hybrid loss function is one of the keys on how to improve specific characteristics of an image. Now, comparing the benchmark models with RED-GAN. The concept of using GAN with RED-CNN (PSNR/SSIM = 37.59/0.9102) surpassed the results of simply using RED-CNN (32.16/0.7315) or when using RED-CNN with self attention modules (32.33/0.8015) but comparatively almost the same as the DRL-E model (37.42/0.8904). That being said, this experiment is deemed successful for showing the effectiveness of deep learning technique, specifically using a GAN architecture, for LDCT image denoising.

8 Conclusion

Computed tomography is a non-invasive modality tool for diagnosing abnormalities inside the human body. The use of X-ray radiation in commercial CT scanners can cause harm to the patient overtime. With the booming of deep learning, this radiation dose can be reduced without affecting the quality of medical image. The

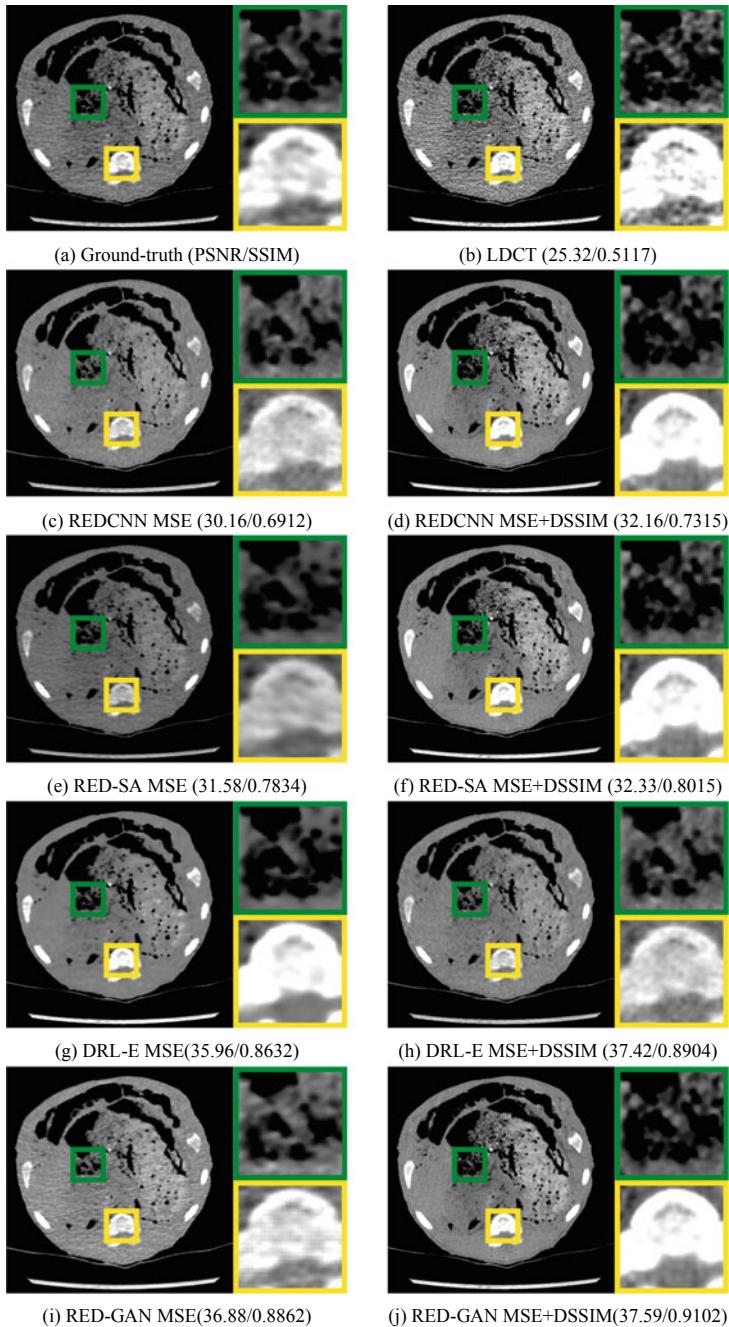


Fig. 7 Visual and quantitative results

simple experiment done demonstrated the potential clinical integration of using GAN denoising algorithms and can be improved especially with the fast advancement of technology.

References

1. Buzug, T. M. (2008). *Computed tomography: From photon statistics to modern cone-beam CT*. Springer Berlin/Heidelberg [Online]. Available <http://ebookcentral.proquest.com/lib/ryerson/detail.action?docID=372096>
2. Mettler, F. A., Huda, W., Yoshizumi, T. T., & Mahesh, M. (2008). Effective doses in radiology and diagnostic nuclear medicine: a catalog. *Radiology*, 248(1), 254–263, pMID: 18566177 [Online]. Available: <https://doi.org/10.1148/radiol.2481071451>
3. Kalender, W. A. (2006). X-ray computed tomography. *Physics in Medicine & Biology*, 51(13), R29–R43 [Online]. Available: <https://doi.org/10.1088/0031-9155/51/13/r03>
4. Cadth optimal use report: The canadian medical imaging inventory, 2015. *Canadian Medical Inventory*, Jan 2021 [Online]. Available: <https://www.cadth.ca/executive-summary-cmii-2019-2020>
5. Medical technologies. (2019). *Health at a glance 2019: OECD indicators* (p. 243). [Online]. Available: <https://doi.org/10.1787/4dd50c09-en>
6. Marcos, L. (2022). *Fused attention modules embedded in artificial neural networks for low dose ct denoising with integrated loss functions*. M.S. thesis, Dept. of Electrical and Computer Engineering, Toronto Metropolitan University, Toronto, ON, Canada.
7. Kalra, A. (2018). Developing fe human models from medical images. In K.-H. Yang (Ed.) *Basic finite element method as applied to injury biomechanics* (pp. 389–415). Academic Press. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B978012809831800009X>
8. Zhou, H., Tamura, T., Kusaka, Y., Suganuma, N., Subhannachart, P., Vijitsanguan, C., Noisiri, W., Hering, K. G., Akira, M., Itoh, H., Arakawa, H., Ishikawa, Y., Kumagai, S., & Kurumatani, N. (2012). Development of a guideline on reading CT images of malignant pleural mesothelioma and selection of the reference CT films. *European Journal of Radiology*, 81(12), 4203–4210, imaging in Acute Chest Pain. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0720048X12003920>
9. Lev, M., & Gonzalez, R. (2002). ct angiography and ct perfusion imaging. In A. W. Toga, & J. C. Mazziotta (Eds.) *Brain mapping: The methods* (2nd ed., pp. 427–484). Academic Press. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780126930191500198>
10. Mudgal, P., & Bell, D. J. (2020). X-ray production. In D. J. Bell (Ed.) *Imaging Technology*. Radiopaedia.org. [Online]. Available: <https://radiopaedia.org/articles/25428>
11. Rontgen, W. C. (1896). On a new kind of rays. *Science*, 3(59), 227–231. [Online]. Available: <https://www.science.org/doi/abs/10.1126/science.3.59.227>
12. Smith-Bindman, R., Lipson, J., Marcus, R., Kim, K. P., Mahesh, M., Gould, R., Berrington de Gonzalez, A., & Miglioretti, D. L. (2009). Radiation dose associated with common computed tomography examinations and the associated lifetime attributable risk of cancer. *Archives of Internal Medicine*, 169(22), 2078–2086. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/20008690/>
13. De Gonzalez, A. B., Mahesh, M., Kim, K. P., Bhargavan, M., Lewis, R., Mettler, F., & Land, C. (2009). Projected cancer risks from computed tomographic scans performed in the United States in 2007. *Archives of Internal Medicine*, 169(22), 2071–2077. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/20008689/>
14. Martinez, N., Wieder, J., & Schneider, T. (2021). *Tg 114: The three r's of reasonable: Relationships, rationale, and resources*. International Commission on Radiological Protection (ICRP),

- Clemson University, USA and Environmental Protection Agency, USA and CEPN, France. [Online]. Available: <https://www.icrp.org/page.asp?id=520>
- 15. Kulathilake, K. S. H., Abdullah, N. A., Sabri, A. Q. M., Bandara, A. R., & Lai, K. W. (2022). A review on self-adaptation approaches and techniques in medical image denoising algorithms. *Multimedia Tools and Applications*, 1218, 1–36. [Online]. Available: <https://doi.org/10.1007/s11042-022-13511-w>
 - 16. Diwakar, M., & Kumar, M. (2018). A review on CT image and its denoising. *Biomedical Signal Processing and Control*, 42, 73–88. [Online]. Available: <https://doi.org/10.1016/j.bspc.2018.01.010>
 - 17. Lee, M. S., Park, S. W., Lee, S. Y., & Kang, M. G. (2017). Motion-adaptive 3d nonlocal means filter based on stochastic distance for low-dose x-ray fluoroscopy. *Biomedical Signal Processing and Control*, 3, 74–85. [Online]. Available: <https://doi.org/10.1016/j.bspc.2017.05.001>
 - 18. Kumar, M., & Mishra, S. K. (2015). Particle swarm optimization-based functional link artificial neural network for medical image denoising. In *Computational vision and robotics* (pp. 105–111). Springer India. [Online]. Available: https://doi.org/10.1007/978-81-322-2196-8_13
 - 19. Trung, N. T., Trinh, D.-H., Trung, N. L., & Luong, M. (2022). Low-dose ct image denoising using deep convolutional neural networks with extended receptive fields. *Signal, Image and Video Processing*. [Online]. Available: <https://doi.org/10.1007/s11760-022-02157-8>
 - 20. Marcos, L., Alirezaie, J., & Babyn, P. (2022). Low dose CT denoising by resnet with fused attention modules and integrated loss functions. *Frontiers in Signal Processing*, 1. [Online]. Available: <https://doi.org/10.3389/frip.2021.812193>
 - 21. Gholizadeh-Ansari, M., Alirezaie, J., & Babyn, P. (2019). Deep learning for low-dose ct denoising using perceptual loss and edge detection layer. *Journal of Digital Imaging*, 33(2), 505–514. [Online]. Available: <https://doi.org/10.1007/s10278-019-00274-4>
 - 22. Han, Z., Shangguan, H., Zhang, X., Zhang, P., Cui, X., & Ren, H. (2022). A dual-encoder-single-decoder based low-dose ct denoising network. *IEEE Journal of Biomedical and Health Informatics*, 26(7), 3251–3260. [Online]. Available: <https://doi.org/10.1109/JBHI.2022.3155788>
 - 23. Marcos, L., Alirezaie, J., & Baby n,P. (2021). Low dose CT image denoising using boosting attention fusion gan with perceptual loss. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)* (pp. 3407–3410). [Online]. Available: <https://doi.org/10.1109/EMBC46164.2021.9630790>
 - 24. Yi, X., & Babyn, P. (2018). Sharpness-aware low dose CT denoising using conditional generative adversarial network. *Journal of Digital Imaging*, 31, 655–669.
 - 25. Jing, J., Xia, W., Hou, M., Chen, H., Liu, Y., Zhou, J., & Zhang, Y. (2022). Training low dose CT denoising network without high quality reference data. *Physics in Medicine & Biology*, 67(8), 084002. [Online]. Available: <https://doi.org/10.1088/1361-6560/ac5f70>

Generating 3D Reconstructions Using Generative Models



Mehdi Malah, Ramzi Agaba, and Fayçal Abbas

Abstract As the capacity for visual representation continues to evolve, there is a growing need for techniques for realistic and efficient creation of three-dimensional objects. Generative models, particularly Generative Adversarial Networks, Variational Autoencoders and novel methods of Text-to-3D, utilize textual descriptions to generate 3D reconstructions with high-quality geometry and disentangled materials. In this chapter, we present an in-depth exploration of the application of generative models in 3D reconstruction. We begin by discussing the theoretical underpinnings of these models and their applicability to 3D reconstruction. This chapter studies how these models learn to generate new instances from a given distribution. We end with discussions of potential future directions and the broader impacts of these technologies in various industries.

Keywords Generative models · 3D reconstruction · 3D modeling · Mesh generation · Synthetic data generation

M. Malah (✉)

ICOSI Laboratory, Computer Science Department, University of Abbes Laghrour, Khencela, Algeria

e-mail: malah.mehdi@univ-khenchela.dz

R. Agaba

ReLaCS2 Laboratory, Computer Science Department, University Larbi Ben Mhidi, Oum El Bouaghi, Algeria

e-mail: ramzi.agaba@univ-oeb.dz

F. Abbas

LESLA Laboratory, Computer Science Department, University of Abbes Laghrour, Khencela, Algeria

e-mail: abbas_faycal@univ-khenchela.dz

1 Introduction

Three-dimensional (3D) reconstruction and generative models are rapidly expanding fields of research that have seen significant advances in recent years. Both areas play a crucial role in a variety of applications, including computer vision, robotics, medical imaging, and computer graphics.

3D reconstruction involves the process of capturing the geometric structure of objects or scenes from a set of images or other sensory data [1]. The field has made substantial progress with the development of novel algorithms and techniques, such as structure from motion (SfM) [2], multi-view stereo (MVS) [1], and depth estimation from single images [3]. The rise of deep learning methods has also led to advancements in 3D reconstruction, including the utilization of convolutional neural networks and generative adversarial networks (GANs) [4].

Generative models, on the other hand, are a class of machine learning algorithms that learn to generate new data samples similar to a given dataset. Two popular types of generative models are Variational Autoencoders (VAEs) [5] and GANs [4]. These models have shown impressive results in generating realistic 2D images [6], as well as 3D objects and scenes [7, 8]. Moreover, the integration of 3D reconstruction techniques and generative models has demonstrated potential for applications such as shape completion [9, 10] and scene synthesis [11, 12].

3D reconstruction is an essential task in computer vision and graphics, aiming to recover the three-dimensional structure of objects or scenes from a set of 2D images or depth maps [13–15]. Traditional methods, such as multi-view stereo or structure-from-motion, have limitations in terms of robustness, scalability, and accuracy, especially when dealing with complex, noisy, or incomplete data. Generative models, particularly deep learning-based approaches, have emerged as a promising alternative for addressing these challenges.

Generative models can learn to handle noisy, occluded, or missing data by leveraging their capacity to learn complex distributions and capture global and local dependencies in the input data. Generative models can be trained on large-scale datasets, allowing them to capture a wide range of object and scene structures. This scalability enables the generation of realistic and diverse 3D reconstructions, addressing the limitations of traditional methods that often struggle with large or complex data. Generative models can be adapted for various tasks, such as single-view or multi-view 3D reconstruction, depth estimation, or even semantic segmentation. Additionally, they can be fine-tuned for specific domains or applications, making them a versatile solution for different 3D reconstruction problems. These models have been successfully applied to various domains, demonstrating their potential for addressing complex 3D reconstruction problems.

Fig. 1 Example of point cloud representation



2 Background

2.1 3D Data Representation

Three-dimensional data representation is a critical aspect of computer graphics and computer vision applications, as it enables the accurate and efficient storage and manipulation of 3D models. There are three main types of 3D data representation: point cloud, mesh, and voxel. Each representation has its own advantages and limitations depending on the specific application.

2.1.1 Point Cloud Representation

A point cloud is a collection of 3D points that represent the surface of an object. Each point is typically defined by its x, y, and z coordinates, and may also have additional information such as color, normal vector, or intensity. Point cloud data is commonly acquired using 3D scanners, lidar systems, or photogrammetry techniques. One of the main advantages of point cloud data is that it can accurately represent the surface details of complex objects. However, point clouds can be computationally expensive to process, and they may require significant storage space to store large datasets. See Fig. 1, which represents an example concerning the point cloud representation.

2.1.2 Mesh Representation

A mesh is a collection of interconnected polygons that form a 3D surface. Each polygon is defined by its vertices and may also have additional information such as texture coordinates or material properties. Meshes are widely employed in computer graphics applications such as video games or 3D modeling software. One of the main advantages of meshes is that they can be efficiently rendered and manipulated by graphics hardware. However, meshes can be challenging to create and modify,

Fig. 2 Example of mesh representation

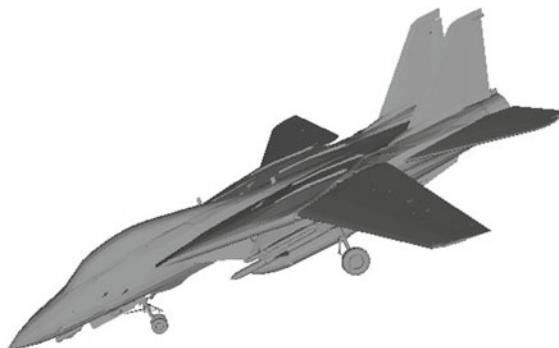
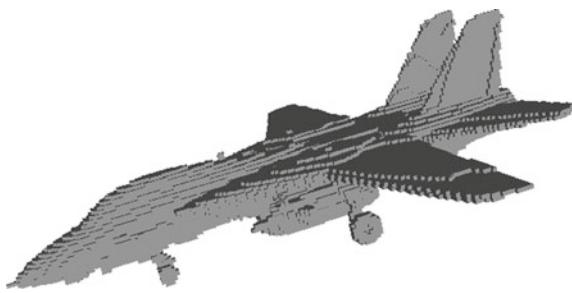


Fig. 3 Example of voxel representation



particularly for complex shapes with many polygonal faces [16]. See Fig. 2, which represents an example concerning the mesh representation.

2.1.3 Voxel Representation

A voxel is a 3D pixel that represents a volume element. Each voxel is typically defined by its location in 3D space and a value that represents a property such as color or density. Voxel data is commonly used in medical imaging applications such as CT or MRI scans. One of the main advantages of voxel data is that it can accurately represent the internal structures of objects. However, voxel data can be computationally expensive to process and may require significant storage space to store large datasets. See Fig. 3, which represents an example concerning the voxel representation.

Each 3D data representation has its own strengths and limitations. Point clouds are suitable for representing detailed surface information, meshes are useful for efficient rendering and manipulation, and voxels are ideal for representing internal structures. The choice of representation depends on the specific requirements of the application, including computational resources, storage space, and accuracy.

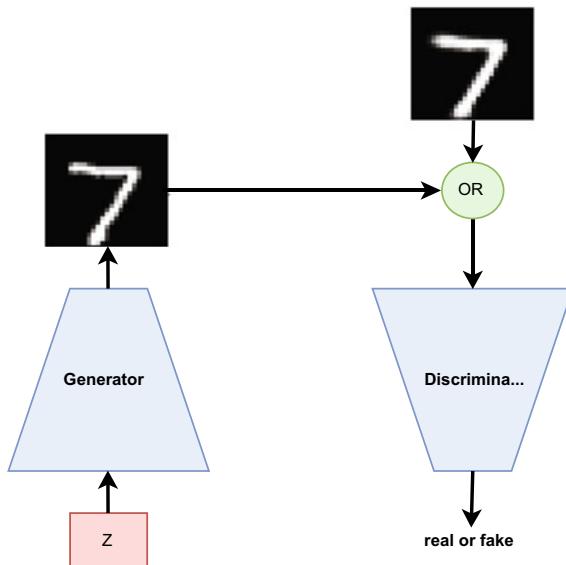


Fig. 4 Generative adversarial network (GAN) architecture

2.2 Generative Models

Generative models are a class of machine learning algorithms that are designed to generate new data samples similar to the training data. They have been widely used in various applications, such as image and text generation, data augmentation, and data compression. Generative models have shown impressive results in generating realistic and high-quality data, which makes them attractive for many practical applications. One of the most popular generative models is the Generative Adversarial Network introduced by Kingma and Welling [5]. GANs are composed of two neural networks: a generator network that generates false samples and a discriminator network that attempts to differentiate between fake and actual samples (See Fig. 4). In a game-theoretic framework, the two networks are jointly trained, with the generator network attempting to generate samples that deceive the discriminator network and the discriminator network attempting to correctly identify the false samples. GANs have been applied to various domains such as image synthesis, video generation, and text-to-image synthesis.

VAE, which was introduced by Kingma and Welling [5], is another prominent class of generative models. Encoder networks map input data to a lower-dimensional representation, while decoder networks map the lower-dimensional representation back to the original data space. VAEs are trained by optimising a variational lower bound on the likelihood of the data, which encourages the encoder to learn a meaningful latent representation that encapsulates the data's underlying structure (See Fig. 5).

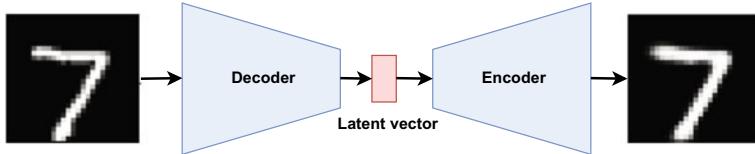


Fig. 5 Variational autoencoders (VAE) architecture

have been applied to various domains such as image generation, text generation, and speech synthesis.

The goal of generative modeling is to model the probability distribution of the input data, typically denoted by $P(X)$, where X is the random variable representing the data. The generative model is trained on a dataset, D , consisting of N samples, x_1, x_2, \dots, x_n drawn independently from $P(X)$.

The model is typically parametrized by a set of learnable parameters, Θ , which are optimized to maximize the log-likelihood of the training data:

$$\Theta^* = \operatorname{argmax}_{\Theta} \sum \log P(x_i | \Theta) \quad (1)$$

where $P(x_i | \Theta)$ is the likelihood of sample x_i under the model parameterized by Θ . The log-likelihood is used instead of the likelihood to avoid numerical underflow when working with small probabilities.

One common type of generative model is the autoregressive model, which models the joint probability of the input data as a product of conditional probabilities:

$$P(X) = \prod_i P(x_i | x_1, x_2, \dots, x_{i-1}) \quad (2)$$

where $P(x_i | x_1, x_2, \dots, x_{i-1})$ is the conditional probability of x_i given the previous inputs. This allows the model to generate new samples by iteratively sampling from the conditional distributions, starting with an initial value for x_1 .

3 3D Reconstruction Using Generative Models

3D reconstruction is a vital component in various applications such as computer-aided design, robotics, and virtual reality. In recent years, generative models have shown remarkable capabilities in reconstructing 3D objects from various input data types, such as 2D images, point clouds, and depth maps. Three-dimensional reconstruction has gained significant attention due to its potential in various applications. Generative models have emerged as a promising approach to tackle this problem, leveraging their ability to learn complex data distributions and generate realistic 3D objects.

3.1 Autoencoders for 3D Reconstruction

Pintelas and Pintelas [17] introduces a novel deep learning framework for 3D object recognition and generation. The authors propose a 3D Convolutional AutoEncoder (3D-CAE) to learn a compact representation of 3D objects in a hierarchical and unsupervised manner. The 3D objects are represented as volumetric grids of occupancy values, allowing the model to process 3D shapes directly. The 3D-CAE architecture consists of an encoder and a decoder (see Fig. 6).

The encoder is a 3D CNN that compresses the volumetric input into a lower-dimensional latent space. The decoder is another 3D CNN that reconstructs the input object from the latent representation. The training objective is to minimize the reconstruction error between the input and the reconstructed object. The generative capabilities of the model enable the synthesis of novel 3D shapes and the completion of partially observed objects.

Yang et al. [18] introduced the Point Cloud AutoEncoder, a deep learning framework designed for 3D point cloud data. The authors propose a novel autoencoder architecture that can effectively learn latent representations of 3D point clouds, while also being able to generate new point cloud samples. The Point Cloud AutoEncoder consists of an encoder and a decoder, with the encoder extracting a global feature vector from the input point cloud and the decoder reconstructing the input from this lower-dimensional representation.

The key innovation in the proposed model is the use of the Chamfer pseudo-distance as the loss function, which measures the similarity between the input and reconstructed point clouds. The authors demonstrate the effectiveness of their model on various tasks, including point cloud autoencoding, generation, and manipulation. They show that the PointNet AutoEncoder outperforms other state-of-the-art methods on these tasks, particularly in terms of reconstruction quality and latent space interpretability.

The Point Cloud AutoEncoder consists of an encoder and a decoder, with the encoder extracting a global feature vector from the input point cloud and the decoder reconstructing the input from this lower-dimensional representation. The key innovation in the proposed model is the use of the Chamfer pseudo-distance as the loss function, which measures the similarity between the input and reconstructed point clouds. The authors demonstrate the effectiveness of their model on various tasks, including point cloud autoencoding, generation, and manipulation. They show that the PointNet AutoEncoder outperforms other state-of-the-art methods on these tasks, particularly in terms of reconstruction quality and latent space interpretability.

Dai et al. [19] presented a novel deep learning framework called the 3D-Encoder-Predictor Network (3D-EPN) for the task of volumetric 3D object completion see the overall architecture in Fig. 7. The main goal of this paper was to tackle the challenge of reconstructing complete 3D shapes from partial or incomplete observations. The 3D-EPN framework consists of two main components: a 3D convolutional autoencoder and a 3D predictor. The 3D convolutional autoencoder is responsible for

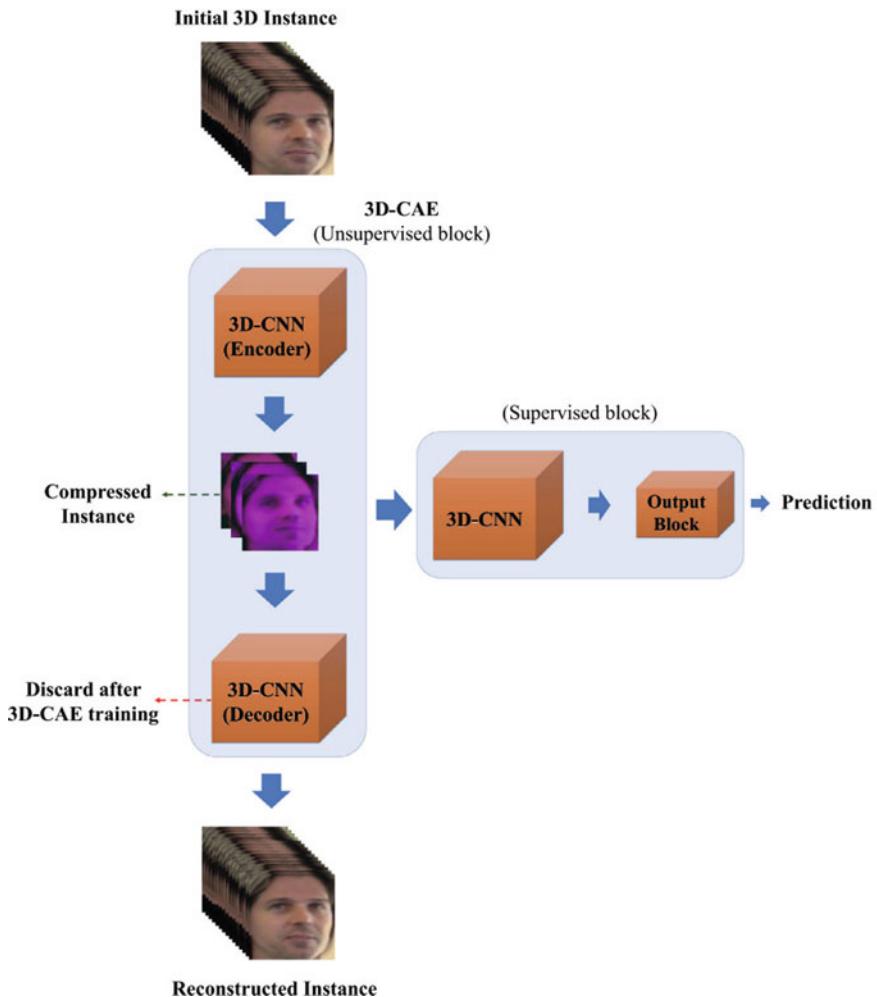


Fig. 6 3D convolutional AutoEncoder (3D-CAE) architecture [17]

learning a compact, low-dimensional representation of complete 3D shapes, while the 3D predictor is tasked with mapping the partial input data to the learned low-dimensional representation. In their experiments, the authors trained their model on a dataset of complete 3D shapes and demonstrated that the 3D-EPN can effectively complete a wide range of partially observed objects. They also showed that their model outperforms other state-of-the-art methods in terms of both quantitative and qualitative evaluations.

3.2 Variational Autoencoders for 3D Reconstruction

VAEs introduce a probabilistic framework to autoencoders, allowing them to generate diverse and realistic 3D reconstructions [5]. Wu et al. [7] introduced the 3D-VAE-GAN, which combines VAEs and Generative Adversarial Networks for generating 3D object representations. The model architecture comprises a 3D VAE as the generator and a 3D CNN as the discriminator. The 3D-VAE-GAN has demonstrated success in synthesizing 3D objects from 2D images and interpolating between object shapes.

Rezende et al. [20] proposed the 3D VAE-Expectation Maximization (VAE-EM) model to reconstruct 3D shapes from 2D images. The model utilizes the Expectation Maximization (EM) algorithm to iteratively refine the 3D shape reconstruction. The 3D VAE-EM has been shown to outperform previous methods in terms of reconstruction quality and convergence speed.

Groueix et al. [21] introduced AtlasNet-VAE, a method for 3D shape reconstruction from single-view images. The AtlasNet-VAE architecture consists of a VAE that learns to generate 2D parameterized charts, which are then mapped onto 3D surfaces using a learned deformation function. The model has been demonstrated to produce high-quality 3D reconstructions, even when trained on limited data.

Occupancy Networks (ONet) were introduced by Mescheder et al. [22] as a novel and flexible representation of 3D geometry, which have found applications in various areas such as computer graphics, robotics, and computer vision. ONet is a deep learning framework that represents a 3D shape as a continuous function over the 3D Euclidean space, effectively enabling the reconstruction of complex geometries with arbitrary topologies.

ONet combines the advantages of both implicit and explicit representations, such as voxel grids and point clouds, by providing an efficient and expressive representation of 3D geometry.

3.2.1 GAN-based Models for 3D Reconstruction

Generative Adversarial Networks have shown promising results in various computer vision applications, such as image synthesis, style transfer, and inpainting. In recent years, GAN-based models have been applied to 3D reconstruction tasks, generating

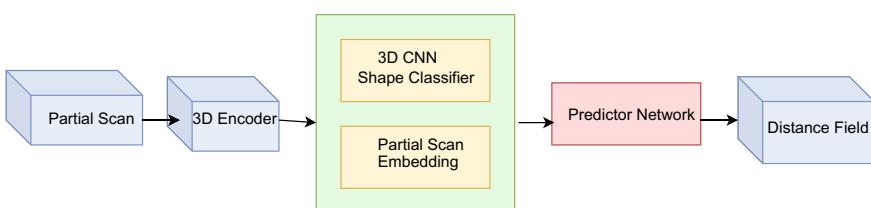


Fig. 7 Architecture of 3D encoder-predictor network

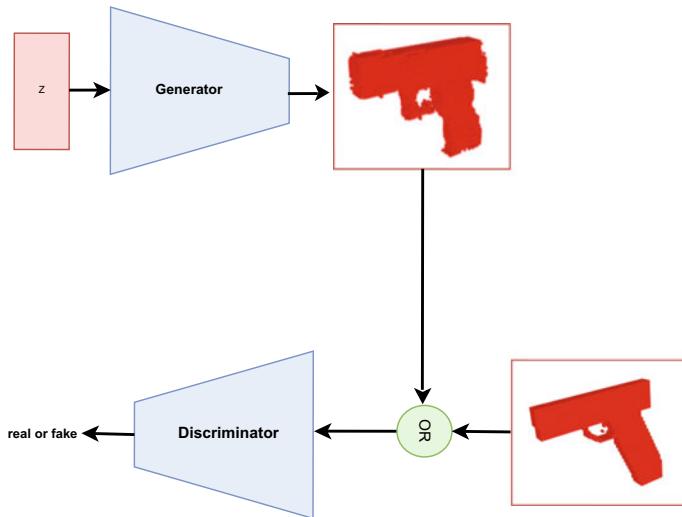


Fig. 8 Generator architecture in [7]

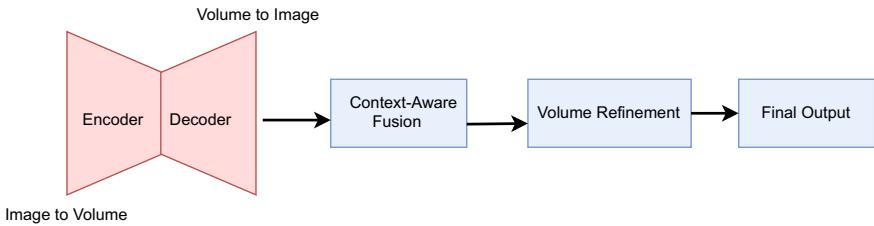


Fig. 9 The overview of the model Pix2Vox

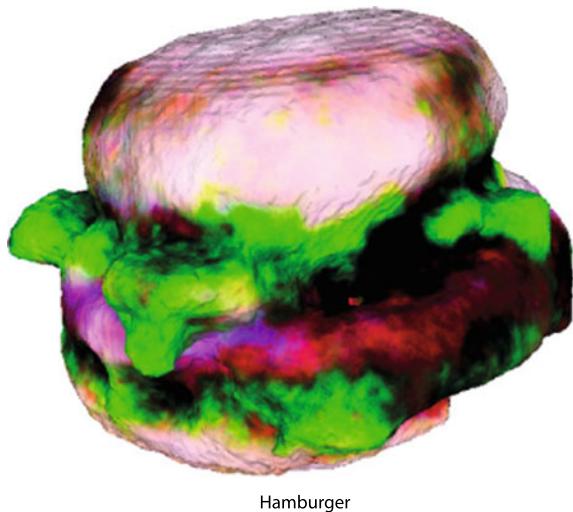
detailed and accurate representations of objects from 2D images or partial 3D data. This section provides an in-depth analysis of GAN-based models that have been specifically designed for 3D reconstruction.

Wu et al. [7] proposed a GAN architecture called 3D-GAN for generating 3D objects from random noise. The 3D-GAN is comprised of a generator and discriminator network that operate on voxel-based 3D representations. This model has demonstrated the ability to generate diverse and visually appealing 3D objects while maintaining high quality and fidelity. Figure 8 represents the GAN architecture used.

Pix2Vox is a framework proposed by Xie et al. [23] that reconstructs 3D objects from 2D images. Figure 9 represents the different stages of the 3D reconstruction. The image encoder extracts features from 2D images, which are then used by the 3D generator to create voxel-based 3D representations. The 3D discriminator assesses the quality of the generated 3D objects. Pix2Vox has demonstrated robust performance in 3D reconstruction tasks, even with complex shapes and structures.

Liu et al. [24] developed the Point Set Generative Adversarial Network (PSGAN) for 3D object reconstruction from a single image. The PSGAN architecture consists

Fig. 10 Example generated by the DreamFusion method



of a generator that predicts point sets and a discriminator that evaluates the quality of the generated point sets. The model leverages Chamfer distance-based loss and adversarial loss to produce high-quality 3D reconstructions. The PSGAN has demonstrated the ability to generate detailed and accurate point-cloud.

3.2.2 Text-to-3D Models for 3D Reconstruction

In recent years, there has been a growing interest in the development of text-to-3D methods, which aim to generate 3D models from natural language descriptions. The primary motivation for this research area is to enable non-expert users to create 3D content using simple textual input. This has applications in fields such as computer-aided design, video game design, virtual reality, and more. One of the earliest approaches to text-to-3D is rule-based methods, which rely on predefined rules or templates to convert textual input into 3D models.

An example of this approach is the work by Coyne and Sproat [25], who used a rule-based system to generate 3D architectural models from text descriptions. However, these methods face limitations in scalability and adaptability to new input scenarios.

Poole et al. [26] present the model DreamFusion(see Fig. 10). The fundamental concept behind DreamFusion is the translation of text descriptions into 2D diffusion patterns, which are then processed and translated into 3D object models. By utilizing the diffusion process, the system can effectively represent complex structures in a highly flexible and scalable manner. The authors introduced a novel framework that leverages state-of-the-art language models and diffusion models. The language models are used to interpret the textual descriptions and convert them into a form suitable



Fig. 11 High-definition geometric objects with materials generated by Chen et al. [27]

for the diffusion model. The diffusion model then takes these representations and generates the corresponding 2D patterns. The paper reports significant success in generating accurate, high-quality 3D models from a diverse range of textual descriptions. The authors also present an extensive evaluation of their system, demonstrating its superior performance compared to existing methods in terms of both quality and versatility.

The central contribution of Fantasia3D [27] is its ability to disentangle the aspects of geometry and appearance in the 3D content creation process. By separating these two components, the system can generate more nuanced and accurate 3D models based on text inputs. Geometry relates to the shape and structure of the model, while appearance encompasses aspects like texture, color, and material. The model is designed with a two-step approach. Firstly, a geometric model is created based on the textual description.

This model, although devoid of any specific appearance attributes, captures the general shape and structure as described in the text. In the second step, appearance attributes are added to the geometric model, again based on the text input. The authors trained Fantasia3D using a large-scale dataset consisting of 3D models alongside their corresponding textual descriptions. The model was trained to minimize the discrepancy between the generated model and the actual model in the dataset. Evaluations have shown that Fantasia3D outperforms previous methods in terms of the

quality and accuracy of generated 3D content. The results generated by this method in Fig. 11.

4 Evaluating the Quality of 3D Reconstructions

Evaluating the quality of 3D reconstructions generated using generative models is an important aspect of research in this domain. A variety of quantitative and qualitative metrics can be used to assess the quality of generated 3D models. In this discussion, we review some of the commonly used evaluation metrics and methods.

4.1 Chamfer Distance

Chamfer distance (CD) is a commonly used metric for measuring the similarity between two point clouds P and Q, making it suitable for evaluating 3D reconstructions [28]. CD computes the average distance between each point in the generated point cloud and its nearest neighbor in the ground-truth point cloud, and vice versa. A lower CD indicates a better reconstruction. It measures the average distance between each point in P to its nearest neighbor in Q, and vice versa. The Chamfer distance is defined as:

$$CD(P, Q) = \frac{1}{|P|} \sum_{p \in P} \min_{q \in Q} \|p - q\|^2 + \frac{1}{|Q|} \sum_{q \in Q} \min_{p \in P} \|q - p\|^2 \quad (3)$$

where $|P|$ and $|Q|$ are the cardinalities of point clouds P and Q, respectively, and $\|\cdot\|$ denotes the Euclidean distance.

4.2 Earth Mover's Distance

Earth Mover's Distance (EMD) is another popular metric for evaluating 3D reconstruction quality by measuring the dissimilarity between two point clouds P and Q [29]. EMD calculates the minimum cost required to transform one point cloud into another, considering the pairwise distances between points. A lower EMD indicates better reconstruction quality. EMD can be calculated by solving the following optimization problem:

$$EMD(P, Q) = \min_{\Phi: P \rightarrow Q} \sum_{p \in P} \|p - \Phi(p)\|^2 \quad (4)$$

where Φ is a bijection between P and Q.

4.3 Intersection Over Union (IoU)

Intersection over Union is a widely used metric for measuring the similarity between two 3D volumes, such as voxel grids or meshes. IoU is calculated as the ratio of the volume of intersection between the reconstructed and ground-truth volumes to their volume of union. A higher IoU value indicates better reconstruction quality.

$$IoU = \frac{\text{Volume}(\text{Reconstruction} \cap \text{GroundTruth})}{\text{Volume}(\text{Reconstruction} \cup \text{GroundTruth})} \quad (5)$$

Each of these evaluation metrics has its advantages and drawbacks. Chamfer distance is computationally efficient and is robust to different point cloud densities. However, it may not capture fine-grained differences between reconstructions and ground truth, as it considers only pairwise point distances.

Earth Mover’s Distance is more sensitive to local structure differences but is computationally expensive, making it less suitable for large-scale evaluations. Additionally, EMD assumes a bijection between the point clouds, which may not always be valid. Intersection over Union is a simple and intuitive measure, but it requires voxelization of the 3D models, which may introduce discretization errors. Moreover, IoU is sensitive to misalignments between the reconstructions and ground truth.

5 Future Directions

The field of 3D reconstruction using generative models has witnessed significant advancements in recent years. However, there are still many challenges and opportunities for further research. Here, we provide an overview of potential future directions for research in this field:

Enhanced accuracy and resolution Improving the accuracy and resolution of 3D reconstructions is a crucial research goal. This can be achieved by developing novel generative models that capture finer details and more realistic object shapes. Additionally, incorporating multi-scale or hierarchical representations within the models can help capture the different levels of detail present in real-world objects.

Efficient and real-time reconstruction Developing efficient generative models that enable real-time 3D reconstruction is essential for various applications, such as robotics and augmented reality. This can be achieved through model optimization, hardware acceleration, and leveraging advanced algorithms that exploit the structure and sparsity of 3D data.

Handling occlusions and missing data Occlusions and missing data are common challenges in 3D reconstruction tasks. Developing generative models that can robustly handle these issues and accurately reconstruct the underlying 3D structures is an important research direction. Techniques such as data augmentation, self-

supervised learning, or incorporating prior knowledge about the 3D structures can help address these challenges.

Incorporating additional modalities Integrating additional data modalities, such as depth, surface normals, or semantic information, can improve the quality of 3D reconstructions. Future research can explore novel methods for fusing different data types and leveraging their complementary information to enhance the generative models' performance.

Unsupervised and self-supervised learning Exploring unsupervised and self-supervised learning techniques for 3D reconstruction can help reduce the dependence on large annotated datasets. These techniques can enable models to learn useful representations from unlabeled data or exploit the inherent structure of the 3D data for learning.

Robustness and generalization Developing generative models that exhibit robustness and generalization across different object categories, viewpoints, and lighting conditions is a critical research goal. This can be achieved through the development of more expressive and flexible model architectures, as well as leveraging techniques such as domain adaptation or transfer learning.

Application-specific 3D reconstruction There is potential for research on application-specific 3D reconstruction methods that cater to the unique requirements of different domains, such as medical imaging, robotics, or cultural heritage preservation. Tailoring generative models for these specific applications can lead to significant improvements in reconstruction quality and utility.

Evaluation and benchmarking Establishing standardized evaluation protocols and benchmarks for 3D reconstruction tasks is essential for the fair comparison of different methods and the identification of promising research directions. Future research can focus on developing more comprehensive evaluation metrics and benchmark datasets that capture the diverse challenges and requirements of real-world 3D reconstruction tasks.

Application of Transformer Models Transformer models, such as BERT, GPT-4, and their successors, have shown great promise in NLP. A future direction might involve applying these architectures to Text-to-3D reconstruction, which could drastically improve the performance of current models.

6 Conclusion

Generative models have emerged as a powerful tool for 3D reconstruction from a variety of sources, including 2D images, point clouds, and other sensor data. By learning the underlying distribution of the data, generative models can generate new samples that are consistent with the input data and can be used to generate 3D reconstructions. Autoencoders models, variational autoencoders, and generative adversarial networks are all effective generative models for 3D reconstruction, each with its own strengths and weaknesses. Autoregressive models are particularly effective for reconstructing 3D shapes from 2D images, while variational autoencoders are well-suited for gen-

erating 3D models from point clouds. Generative adversarial networks, on the other hand, are effective at generating high-quality 3D reconstructions from a variety of sources. Generative models have the potential to revolutionize 3D reconstruction by providing a powerful tool for generating high-quality 3D models from a variety of sources. As research in this area continues to evolve, we can expect to see even more advanced generative models that can generate more accurate and realistic 3D reconstructions, opening up new possibilities for applications in fields such as virtual and augmented reality, robotics, and more.

References

1. Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., & Szeliski, R. (2006, June). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* (Vol. 1, pp. 519–528). IEEE.
2. Ullman, S. (1979). The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 203(1153), 405–426.
3. Eigen, D., Puhrsch, C., & Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems* (Vol. 27).
4. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Bengio, Y. (2014). *Generative adversarial networks*. arXiv preprint [arXiv:1406.2661](https://arxiv.org/abs/1406.2661).
5. Kingma, D. P., & Welling, M. (2013). *Auto-encoding variational bayes*. arXiv preprint [arXiv:1312.6114](https://arxiv.org/abs/1312.6114).
6. Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4401–4410).
7. Wu, J., Zhang, C., Xue, T., Freeman, B., & Tenenbaum, J. (2016). Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in neural information processing systems* (Vol. 29).
8. Gadelha, M., Maji, S., Wang, R. (2017). 3D shape induction from 2D views of multiple objects. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (pp. 4031–4041).
9. Han, X., Li, Z., Huang, H., Kalogerakis, E., & Yu, Y. (2017). High-resolution shape completion using deep neural networks for global structure and local geometry inference. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 85–93).
10. Wen, X., Xiang, P., Han, Z., Cao, Y. P., Wan, P., Zheng, W., & Liu, Y. S. (2022). PMP-Net++: Point cloud completion by transformer-enhanced multi-step point moving paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1), 852–867.
11. Sauer, A., Karras, T., Laine, S., Geiger, A., & Aila, T. (2023). *Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis*. arXiv preprint [arXiv:2301.09515](https://arxiv.org/abs/2301.09515).
12. Tao, M., Bao, B. K., Tang, H., & Xu, C. (2023). *GALIP: Generative adversarial CLIPs for text-to-image synthesis*. arXiv preprint [arXiv:2301.12959](https://arxiv.org/abs/2301.12959).
13. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., & Jiang, Y. G. (2018). Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 52–67).
14. Gkioxari, G., Malik, J., & Johnson, J. (2019). Mesh r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 9785–9795).
15. Malah, M., Hemam, M., & Abbas, F. (2023). 3D face reconstruction from single image with generative adversarial networks. *Journal of King Saud University-Computer and Information Sciences*, 35(1), 250–256.

16. Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., & Ranzuglia, G. (2008, July). Meshlab: an open-source mesh processing tool. In *Eurographics Italian Chapter Conference* (Vol. 2008, pp. 129–136).
17. Pintellos, E., & Pintellos, P. (2022). A 3D-CAE-CNN model for deep representation learning of 3D images. *Engineering Applications of Artificial Intelligence*, 113, 104978.
18. Yang, Y., Feng, C., Shen, Y., & Tian, D. (2018). Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 206–215).
19. Dai, A., Ruizhongtai Qi, C., & Nießner, M. (2017). Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5868–5877).
20. Rezende, D. J., Eslami, S. M., Mohamed, S., Battaglia, P., Jaderberg, M., & Heess, N. (2016). Unsupervised Learning of 3D Structure from Images. In *Advances in neural information processing systems* (pp. 4997–5005).
21. Groueix, T., Fisher, M., Kim, V. G., Russell, B. C., & Aubry, M. (2018). AtlasNet: A Papier-Mâché approach to learning 3D surface generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 216–224).
22. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., & Geiger, A. (2019). Occupancy networks: Learning 3D reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4460–4470).
23. Xie, J., Zheng, Z., Gao, R., Wang, W., Zhu, S. C., & Wu, Y. N. (2019). Pix2Vox: Context-aware 3D reconstruction from single and multi-view images. In *Proceedings of the IEEE International Conference on Computer Vision*. Retrieved from <https://arxiv.org/abs/1901.11153>
24. Liu, Q., Zhou, H., Xu, Q., Liu, X., & Wang, Y. (2020). PSGAN: A generative adversarial network for remote sensing image pan-sharpening. *IEEE Transactions on Geoscience and Remote Sensing*, 59(12), 10227–10242.
25. Coyne, B., & Sproat, R. (2001, August). WordsEye: An automatic text-to-scene conversion system. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (pp. 487–496).
26. Poole, B., Jain, A., Barron, J. T., & Mildenhall, B. (2022). *Dreamfusion: Text-to-3d using 2d diffusion*. arXiv preprint [arXiv:2209.14988](https://arxiv.org/abs/2209.14988).
27. Chen, R., Chen, Y., Jiao, N., & Jia, K. (2023). *Fantasia3D: disentangling geometry and appearance for high-quality text-to-3D content creation*. arXiv preprint [arXiv:2303.13873](https://arxiv.org/abs/2303.13873).
28. Fan, H., Su, H., & Guibas, L. J. (2017). A point set generation network for 3D object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2463–2471).
29. Rubner, Y., Tomasi, C., & Guibas, L. J. (2000). The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2), 99–121.

ChatGPT Implementation in the Metaverse: Towards Another Level of Immersiveness in Education



**Michael Agyemang Adarkwah, Ahmed Tlili, Boulus Shehata,
Ronghuai Huang, Prince Yaw Owusu Amoako, and Huanhuan Wang**

Abstract Novel artificial intelligence (AI) technologies have the capability to reinvent education and enhance the immersive experience of learners. The education industry is continually embracing advanced AI tools to harness their functionalities for optimal teaching and learning. When the metaverse was released, its interactive features were leveraged to promote an engaging learning experience for students. Several applications of the metaverse in schools indicate that metaverse-based education has the potential to provide situated and authentic learning which is highly prized in this twenty-first century. ChatGPT is one such AI tool which is touted to revolutionize education. ChatGPT 3.5 and 4 possess powerful functionalities that have been underscored to provide a fully immersive experience for learners. The quick response rate, level of accuracy, personality, and other abilities of ChatGPT makes it an enabler of metaverse-based education to create a higher level of immersiveness. In this review article, we outline the strengths and weaknesses of both the metaverse and ChatGPT and how ChatGPT can be used to overcome the shortcomings of metaverse-based education. Considerations for implementing ChatGPT in the metaverse are also discussed. It is proposed that educators should integrate ChatGPT into metaverse educational platforms to situate learners in realistic, problem-solving, and creative ways of thinking.

M. A. Adarkwah (✉) · A. Tlili · B. Shehata · R. Huang · H. Wang
Smart Learning Institute of Beijing Normal University, Beijing, China
e-mail: adarkwahmichael1@gmail.com

A. Tlili
e-mail: ahmed.tlili23@yahoo.com

B. Shehata
e-mail: boulus.shehata@gmail.com

R. Huang
e-mail: huangrh@bnu.edu.cn

H. Wang
e-mail: holly.08@live.cn

P. Y. O. Amoako
Nanjing University of Science and Technology, Nanjing, China
e-mail: papaprince@njust.edu.cn

Keywords ChatGPT · Metaverse · Immersiveness · Artificial intelligence · Natural language processing

1 Introduction

There has been a significant forward leap in the application of artificial intelligence (AI) in education as a result of its capabilities. AI simply refers to the increased ability of machines to perform specific tasks and roles peculiar to humans in the workplace and society as a whole [16]. AI technologies include natural language processing (NLP), virtual assistants, robotics, facial recognition, etc. The rise of AI-based solutions has led to a heightened interest in investigating how humans and machines can play a combined role in education. Although some past studies have shown that machines outperform humans [21, 26], it has been underscored that the symbiotic relationship and synergy between humans and machines (human–machine collaboration) lead to improved performance [16, 25].

Human–machine collaboration (HMC) systems have evolved into two main types of technologies involving human-like technology (chatbots) which maintains conversations or promotes communication [24] and digital twin technologies (e.g. metaverse) which possess the ability to create a virtual model of a physical object in the real environment [54]. An integration of the two technologies (chatbots and metaverse) has been proven to create a higher level of immersiveness which is satisfying, comfortable, easy to use, friendly, meaningful, and interesting [48].

The two technologies (chatbots and metaverse) occurring in tandem help create a cyber-physical-social system (CPSS) that facilitates high-quality task execution, in some instances outperforming human potential [61]. Yilma and his colleagues define CPSS as “an environment cohabited by humans and smart devices that are in a virtual and physical interaction” (p. 6). Through HMC, CPSS aid in the design and construction of smart learning environments [66]. CPSS allows the seamless integration of the cyberspace, physical space, and social space in this intelligent era [64] and establishes an effective connection between human intelligence and machine intelligence [62]. It is believed that CPSS will lead to a paradigm shift in the relationship between intelligent systems, humans, and the physical environment [64] by providing a new level of immersiveness [48].

In this study, we provide a literature review on how a recent and advanced chatbots, ChatGPT, can be used to facilitate teaching and learning in the metaverse (see Fig. 1). To do this, we first provide an overview of the concept of metaverse and ChatGPT, provide ten leverage points of educational institutions for using ChatGPT in the metaverse, and conclude with considerations in integrating ChatGPT in the metaverse. Recommendations for future studies are also discussed. To identify systematic steps in integrating the metaverse and ChatGPT, we asked ChatGPT this prompt; “What are the systematic steps to follow to integrate ChatGPT and the metaverse?” Following the guidelines of the American Psychological Association (APA) on how

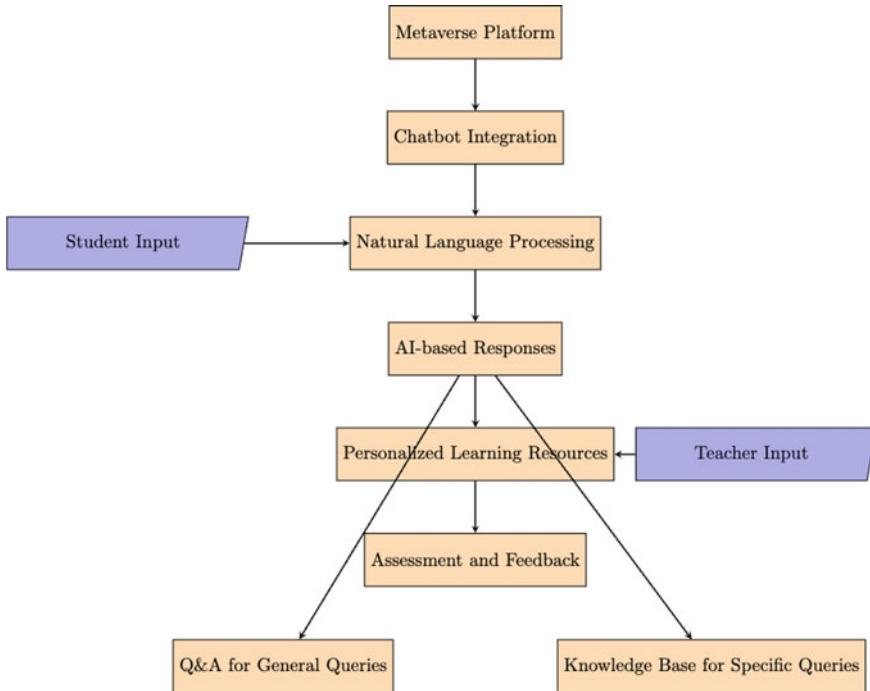


Fig. 1 A flowchart demonstrating how chatbot (ChatGPT) can be integrated into metaverse education

to cite ChatGPT [40], we have included the response of ChatGPT as an appendix (see Appendix).

2 Metaverse

The initial use of the term “metaverse” was by N. Stephenson in his science fiction novel (*Snow Crash*) in 1992 which described an immersive 3D virtual milieu. In the novel, the metaverse is portrayed as a virtual reality (VR) space that makes use of the internet and augmented reality (AR) via avatars and software agents [15]. Although the concept is not new, metaverse received new and massive attention when Facebook was rebranded as “Meta”, which implies a metaverse company. The metaverse is believed to be the next generation of social connection and the internet [19, 27]. Since its inception, educational researchers have applied the term to refer to how learners use digital technologies such as VR, AR, and mixed reality (MR) to engage and interact with one another with avatars [41]. The proliferation of these innovative technologies is accelerating metaverse development.

There was no agreed definition for the term metaverse in literature. Akour et al. [3] depict the metaverse as “an infusion of real and physical universe that allows users to imagine multiple and myriad digital mirrors of the real world, both existent and non-existent, for a variety of purposes” (p. 1). Hwang and Chien [27] also describe it as a created world where people can live under the rules defined by the creator and could be fully or partially virtual.

Over the years, there have been many applications of the metaverse in the field of health, science, business, etc. The metaverse as an alternative medium for learning or communication was accelerated during COVID-19 when everything turned “virtual”. For instance, in the metaverse space, users can collaborate on a project, play games, solve a problem, learn from experiences [27], provide healthcare to patients and medical training to learners [19], and businesses can create virtual products, services, and hospitality and tourism experiences for consumers [23]. Thus, users can use their digital avatars to study, work, play, and communicate with one another [23]. This has led to the creation of educational metaverse platforms to facilitate teaching and learning. Zhang et al. [65] recognized the metaverse as the future of education because of its great features.

3 ChatGPT

A monumental development in artificial intelligence (AI), specifically natural language processing (NLP), has triggered a rise in the creation of powerful chatbots that serves as a gateway for humans to interact with digital devices in such a way that they feel they are communicating with an actual person. The recent popularity of chatbots can be seen in their rapid use in the field of marketing to engage customers in human-like conversations [10, 46], in the health sector to fight infections such as COVID-19 [4], in the field of psychology to provide social and empathetic relationships with users [7], and it is also used to provide entertainment and companionship for the end user [8]. Chatbots have also gained the attention of a growing number of young researchers to conduct conversational surveys [10, 59].

Just to elucidate, “a chatbot or conversational agent (CA) is a software system that can interact or “chat” with a human user in natural language such as English” [28, p. 215]. The conversational agent allows users to communicate and interact with software applications powered by AI-based tools [4]. Educational chatbots (EC) are defined as chatbots developed for explicit learning objectives and are programs that aid in accomplishing an educational or pedagogical goal [32]. Some of the objectives for implementing chatbots in education include improving student skills, increasing the efficiency of education, enhancing students’ motivation and increasing educational access by removing the limitation of physical presence [57]. It has also been underscored that chatbots promote collaborative learning, personalized learning, multimodal communication, scaffolding, and facilitate real-time feedback [32].

ChatGPT is a new viral and advanced chatbot created and made available for public use by OpenAI on November 22. The “chat” refers to a conversational attribute

while the “GPT” stands for Generative Pre-trained Transformer (a machine learning model that understands and generate human-like language). ChatGPT is a generative artificial intelligence (AI) platform trained with a large data set that uses natural language processing (NLP) to respond to users’ text prompts in a realistic manner indistinguishable from a human writer.

In the first week of its release, ChatGPT garnered millions of users worldwide due to the quality of responses to user text prompts on different topics across many disciplines [1, 2]. ChatGPT can write student essays, teleplays and song lyrics, compose music, generate research papers, and answer test questions in just a few seconds [53]. While many people (early adopters) are engaged in suitable interactions with ChatGPT [1], its ethical use and educational benefits are not fully known. In this light, several studies have investigated the potential educational benefits of ChatGPT.

For instance, in their study to see if ChatGPT can be used for finance research, Dowling and Lucey [13] observed that ChatGPT even in its basic state has the potential to generate research studies for reputable journals. They further expounded that with the inclusion of private data and the expertise of researchers, the output from ChatGPT is impressive. Kung et al. [33] investigated the ability of ChatGPT to pass United States Medical Licensing Exam (USMLE). They found that in all three aspects of USMLE, ChatGPT performed at or near the passing score indicating that ChatGPT can assist in medical education. Lund and Wang [38] also added that in educational settings, ChatGPT can be used for literature review assistance, generation of research drafts, analysis of data, language translation, answering questions, and automated summarization. Tlili et al. [51] sum it up by saying that ChatGPT signals a paradigm shift in the educational landscape and other aspects of life.

4 Shortcomings of Metaverse in Education that ChatGPT Can Overcome

Park and Kim [44] mention that the metaverse requires improvement compared to the experience of the real world. For example, there is a need for fast rendering and data analysis in order to process large amounts of data. Matsubara and Oguchi [39] are in consensus that the improvement of response time in metaverse platforms is necessary to ensure its popularity and sustainability. ChatGPT can overcome this challenge because of its capability to generate quick and high-quality responses to users’ text [11, 51]. Also, the avatars in current metaverse platforms have emotional barriers which may cause a user to feel a sense of rejection towards the avatar [44]. ChatGPT can compensate for this deficiency because of its ability to express emotion and personality to some extent [51].

Park and Kim [44] revealed that, although dialogue or conversations in the metaverse are becoming more natural, it is still limited in terms of their ability to model

the multiple personas of humans. That is, there is a need to create an event or environment in the metaverse that can show the diverse persona of users. ChatGPT is able to generate an appropriate response which is indistinguishable from human [38] and has a personality which is able to provide customized and personalized responses to individual users [51]. Rao et al. [47] demonstrated that ChatGPT is able to assess human personalities (personalities of different groups of people) making it appropriate for its application in the metaverse for different users.

An interdisciplinary approach is needed in the design and application of the metaverse because of its real-time changes for a large number of users [44]. ChatGPT has a large data set, hence, is able to provide appropriate responses to diverse topics [51]. Guo and Gao [22] assert that most of the technologies or products embedded in the metaverse are mainly limited to the field of electronic entertainment (e.g. video games) which can lead to a superficial interpretation of what the metaverse means and its potential for pedagogical purposes. In the same light, Kye et al. [34] mentioned that many people perceive the metaverse to be merely an online game environment. ChatGPT is able to engage users in fun, social, and academic activities [12, 51]. Thus, the integration of ChatGPT in the metaverse can enhance educational experiences and potentially alleviate the misconception that the metaverse is a platform solely for entertainment.

How the metaverse can predict real-life unexpected events and outcomes is still a challenge to be overcome [42]. ChatGPT can engage in novel creation [51] and has the capability to make logical predications [5]. When applied in the metaverse, ChatGPT can give consistent predictions of situations in the real world. Current educational applications of the metaverse might cause identity confusion, escape from reality, and maladaptation of real-world students whose identities are yet to be established [34, 49]. ChatGPT constantly reminds users that it is an NLP model and admits its mistakes to alert users that they are interacting with a chatbot and not a human.

Tlili et al. [52] showed that there are pedagogical challenges associated with metaverse in education. According to Tlili et al., the metaverse provides a wide array of digital resources that require teachers to develop competencies. However, because teachers lack these competencies and pedagogical structure, digital resources are not appropriately designed. Additionally, students' insufficient knowledge of the technologies used in the metaverse and how to apply them to read calls for effective time management and continual practice. ChatGPT is simple to use requiring the user not to possess specialized technical skills, ICT competencies or training for its optimal use [51].

To provide a more immersive learning experience in the education sector, Hwang and Chien [27] call for the need to pay attention to curriculum and learning designs in the metaverse such as the use of AI tools that can provide students with immediate feedback and suggestions. ChatGPT can lead to a higher immersive experience by providing real-time and immediate feedback to students tailored to their responses [11, 52].

5 How ChatGPT Can Further Advance Learning in the Metaverse

Through NLP technologies, the responses of ChatGPT can be integrated into virtual assistants or into an avatar of an educational institution in the metaverse to create an immersive learning experience. For instance, through text-to-speech (TTS) technologies, a virtual assistant or an avatar can respond to a learner's inputs in the metaverse using responses from ChatGPT. Learners can obtain meaningful responses to their queries. School leaders or instructional technologists can personalize responses from ChatGPT to each user by integrating user-specific data into the responses. A learner with a specific learning style or an interest in a subject can obtain customized guidance using ChatGPT responses in the metaverse. Through a dialogue management system, the conversational flow between a user and an avatar or a virtual assistant can be smooth and logically coherent. Godwin-Jones [20] mentions that Large language models (LLM) such as ChatGPT will enhance the metaverse features by creating prominent pictures, text, and videos. Below are leverage ten (10) points for educational institutions in the metaverse as a result of the possibility of integrating ChatGPT's responses in a virtual assistant or an avatar.

First, virtual tutoring in the metaverse aided by ChatGPT can be achieved. ChatGPT will be able to understand questions from learners and provide real-time responses. Scholars report the use of conversational agents such as ChatGPT for virtual tutoring in educational settings [37, 56]. For example, in the study by Tellols et al. [50] Sentient Embodied Conversational Agent (SECA) which is able to mimic humanlike sentiment qualities through complex and structured conversations are employed as virtual tutors for children.

Secondly, ChatGPT enhances language learning in the metaverse by providing language learning services. ChatGPT has the ability to adequately translate a speech, provide suggestions on how to improve grammar and pronunciation, and suggest new vocabulary to users. For example, Godwin-Jones [20] LLM like ChatGPT is acknowledged as having the potential to promote language learning in the metaverse.

Thirdly, ChatGPT can provide real-time summaries of a copious text or a full-text document (articles, books, conference papers, reports, etc.) for users to quickly understand a particular concept. In the study by Antaki et al. [5], it was observed that ChatGPT summarizes is on par with traditional fine-tuning methods and is able to perform diverse text summarization tasks.

Also, ChatGPT in the metaverse can be used for assessment purposes through the provision of automated grading and feedback on quizzes, assignments, and examinations. This can help save the time of teachers while at the same time giving learners immediate feedback on their performance. Zhai [63] mentioned that AI tools such as ChatGPT can provide automated grading of assignments and assessments which can reduce the workload of teachers.

For students who are unfamiliar with a particular virtual environment, ChatGPT can be used to help learners navigate virtual libraries and research databases through the provision of the layout and directions on how to access useful resources and

other relevant information. Depending on the educational level and learning style of a learner, ChatGPT can provide book or article recommendations that they might experience difficulty accessing them. Lund and Wang [38] believe that ChatGPT will have a profound impact on academia and libraries in the sense that it has the ability to improve search and discovery, reference and information services, and cataloging.

ChatGPT also possesses to an extent emotion and personality, hence, can be leveraged as an academic advisor or counselor to students in terms of their behavior, course selection, and career decisions in life. Through his user experience, Lin [36] observed that you can ask ChatGPT to be an academic advisor, a therapist, coach, professor, or a tutor. Thus, there are endless possibilities with ChatGPT.

Again, due to the personality of ChatGPT, it can help learners who experience anxiety or emotional distress through the provision of mental support. ChatGPT is able to provide guidance on health and wellness for users to help them live healthy lifestyles [36]. Also, ChatGPT has the potential to build a supportive learning environment for students and provide their psychological needs of relatedness, autonomy, and competence.

ChatGPT can also be an enabler for remote learning in the metaverse. Learners can interact with an avatar of their tutors and obtain educational materials on the metaverse irrespective of time or geographical location. LLM such as ChatGPT can support group and remote learning by providing a structure to facilitate group discussions and automatically generating questions, providing practice problems, assessments, and explanations that are tailored to the needs of individual students so that they can learn at their own pace [29].

Through ChatGPT, real-time collaboration between learners on projects and discussion on course materials is possible in the metaverse. ChatGPT is known to support collaborative writing activities [29]. Learners with their avatars can interact with one another and with the institutional avatar embedded with ChatGPT responses to obtain guidance and relevant tools and resources on how to achieve project goals.

Socialization is also possible through the use of ChatGPT because of its ability to generate fun games, poetry, stories, and songs. This will promote relationship-building in the metaverse. In one study, it was revealed that ChatGPT has the potential to facilitate the exchange of implicit knowledge during socialization by serving as a virtual platform for geographically dispersed teams. This enables team members to share and communicate information with each other regardless of their physical location [31]. Chatbots such as ChatGPT have the potential to act as a companion, friend, or romantic partner for some users [7].

6 Considerations for Implementing ChatGPT in the Metaverse

The metaverse infrastructure or platform should be able to perform language analysis tasks because ChatGPT is based on an NLP model which might require adequate computing resources for it to successfully function. For example, to promote optimal responsiveness, high-performance servers with low latency connections might be needed to ensure ChatGPT is fast in producing outputs. In the metaverse, ensuring low latency among users who are geographically distributed is a practical challenge which must be addressed [9]. Durall and Kapros [14] also mentioned the need to consider the technical aspects of ChatGPT when implementing it. Tlili et al. [51] acknowledged that ChatGPT is more fine-tuned and is trained with a larger dataset which enables it to provide quick response. Notwithstanding, an evaluation of ChatGPT and supporting it with adequate resources for optimal functioning will be needed.

User experience with any novel technology is essential for its rapid use and adoption. In the metaverse, a pilot test on ChatGPT use can be conducted before it is eventually implemented in educational settings for learners. Users need to be aware of the features of ChatGPT to ensure its successful use. ChatGPT should integrate seamlessly with the metaverse interface for users to be able to access its functions. In the study by Tlili et al. [51], a user experience conducted revealed that ChatGPT is able to provide quality responses which are satisfactory in nature but it is not free from mistakes. Participants in the study called for an enhancement of the technology to address such errors or the possibility of ChatGPT being biased. Also, one of the most important objectives of the metaverse is to provide an immersive user experience [9]. Cheng and his colleagues add that in the future, the metaverse will require full-body and high-quality avatars to provide a good immersive experience for users.

There is a need to ensure content moderation when integrating ChatGPT in the metaverse. Contents that are inappropriate depending on the different types of learners in an educational institution should not be included. This helps to ensure a welcoming and safe learning space for all learners. Also, because ChatGPT is able to generate large amounts of data, a data storage plan or policy needs to be considered. For example, it has been reported that there is a possibility for users to use disallowed content when interacting with ChatGPT, hence, the need for a content moderation policy [18]. According to Europol, there's a higher possibility for ChatGPT 4 to not respond to the disallowed content of users and produce more factual information than ChatGPT 3.5. Also, concerns about content moderation in the metaverse have been raised by scholars. Racism and sexual harassment the possibility of disinformation and surveillance are technical issues raised by experts [58]. Thus, it is vital to develop guidelines for governing or regulating content in the metaverse. Kirkpatrick [30] also calls for the need to find the balance between creative freedom and content regulation in the metaverse. According to, unregulated virtual worlds can serve as a haven for inappropriate content such as pornography and hate speech.

Issues relating to inclusivity and accessibility are also key issues when implementing ChatGPT in the metaverse [14, 52]. Learners with different abilities and from

diverse socio-economic backgrounds or locations should be able to access the metaverse space and obtain responses from ChatGPT. For example, learners with special needs might require assistive technologies to navigate the metaverse and interact with ChatGPT. Accessibility is also a technical issue of the metaverse. Although in our contemporary society internet access may not require any specific device, metaverse platforms require users to wear headsets for better interaction in the virtual world which greatly limits accessibility to the metaverse [9]. The lack of accessibility and affordability of headsets and other accessories can affect user participation in the metaverse [17]. Cheng et al. [9] envision that new interface devices which will be developed will promote access to the metaverse without the need for an additional wearable device and that glasses or contact lenses could serve as substitutes for the cumbersome headsets. Also, in designing multicultural metaverse platforms, social and cultural issues should be considered.

User privacy and security should be considered when integrating ChatGPT in the metaverse. Data encryption should be used to ensure user data is not shared with another user in the metaverse. Thus, measures should be put in place to ensure that only ChatGPT has access to user data and there is also no unauthorized access to ChatGPT and user data in the metaverse. Ethical issues of ChatGPT that have been reported by scholars include user privacy and the use of user data [38, 51]. According to Cheng et al. [9], like many online social networks, there are privacy and security issues in the metaverse such as attacks on user authentication and impersonation. That is, the personal information of users can be stolen which compromises their privacy. In integrating the metaverse and ChatGPT, developers and educators need to address this technical issue.

ChatGPT should also be integrated into the metaverse with scalability in mind. That is, as the number of users in the metaverse grows, there might be an increased workload on ChatGPT. It should be designed to handle user tasks by implementing auto-scaling features and load-balancing techniques that can dynamically adjust resources as required. The scalability of ChatGPT has been proven to be good due to its ability to process large volumes of user queries at the same time making it efficient to meet the needs of a large number of users [43]. Notwithstanding, news media such as the Daily Mail reported an occasional crash of ChatGPT on March 20, 2023. For metaverse platforms, scalability is regarded as a technical requirement. For example, Cheng et al. [9] mention that some social VR platforms such as Workrooms and AltspaceVR which are early prototypes of the metaverse experience difficulty in scaling up to ten participants. This is because there is a corresponding increase in uploading and downloading in metaverse platforms as more participants access the platform which can cause the metaverse platform to break down. Hence, recommends advanced networking techniques to guarantee the scalability of the metaverse which requires a huge bandwidth.

Fine-tuning is also required for the metaverse to ensure accuracy and enhance user experience [60]. Bojic [6] add that the fine-tuning of more or less diverse content has the power to provide a real difference to users of online platforms and the metaverse.

Also, to ensure ChatGPT performs effectively in the metaverse, it will need periodic training and fine-tuning. An evaluation of ChatGPT performance in the metaverse is required to incorporate new changes or updates and ensure ongoing model training. Educational institutions might need to fine-tune or customize the responses of ChatGPT to a specific user or group. The current ChatGPT technology is not totally free from mistakes [51]. Radziwill and Benton [45] proposed metrics for the periodic evaluation of intelligent chatbots such as ChatGPT including their efficiency (performance), effectiveness (functionality and humanity), and user satisfaction (affect, ethics and behavior, and accessibility).

There is also an issue of interoperability. Thus, how ChatGPT will be able to interact with other tools, services, and features in the metaverse. Educational institutions should know which technological tool or virtual assistant is compatible with ChatGPT in the metaverse. For the vision of a seamless metaverse to be realized and for users to transverse the metaverse smoothly, there is a need for interoperability standards that enable access from diverse communication systems [35]. The issue of interoperability is considered one of the fundamental challenges that can arise in the metaverse [55]. The authors expound that the heterogeneous nature of hardware implementation of virtual worlds in the large-scale metaverse can result in great interoperability issues.

7 Conclusion

ChatGPT has been shown to improve the level of immersiveness in metaverse-based education in this review. The metaverse is an immersive virtual space that provides opportunities for learners to interact with each other. By leveraging the power of natural language processing and machine learning, ChatGPT can overcome the shortcomings of the metaverse such as the need to improve the response time in metaverse platforms, its limited ability to model the multiple personas of humans, and the challenge of metaverse platforms in predicting real-life events. It was revealed that ChatGPT can facilitate metaverse education in several ways such as virtual tutoring, language translation service (ChatGPT possess a multilingual ability), assessment and feedback, collaboration, summarization, research report writing, social interactions etc. However, there are still technical and ethical issues that need to be monitored when integrating ChatGPT in the metaverse. For example, to ensure the sustainability of ChatGPT in the metaverse, there is a need for adequate infrastructure to ensure its optimal functioning. Also, an evaluation of user experience might be needed from time to time to provide a high level of immersiveness for learners. Ethical issues that need to be tackled include user privacy, user safety, and accessibility. It is recommended that future researchers should explore how ChatGPT can be leveraged in the metaverse for students with special needs. Also, the subscription version of ChatGPT 3.5 which is ChatGPT Plus based on GPT 4 technology might widen the existing digital inequalities in education. It is recommended that policymakers in education and school leaders ensure equity in accessibility to the technology.

Appendix: Steps in Integrating Metaverse and ChatGPT

No.	Step	Explanation
1	Identify the specific use case	Identify the specific use case for integrating ChatGPT and the metaverse. For example, you may want to use ChatGPT to enhance communication and interaction in a virtual classroom or to create more realistic and engaging chatbots for virtual customer service
2	Choose the appropriate platform	Choose the appropriate platform for integrating ChatGPT and the metaverse. Various platforms, such as Unity, Unreal Engine, can be used to create virtual environments and integrate ChatGPT
3	Set up the virtual environment	Set up the virtual environment that will be used for integrating ChatGPT. This may involve creating a virtual classroom or a virtual customer service environment, depending on your specific use case
4	Train the ChatGPT model	Train the ChatGPT model on the specific language data that will be used in the virtual environment. This may involve fine-tuning the model on the specific domain and context of the virtual environment
5	Integrate ChatGPT with the virtual environment	Integrate the ChatGPT model with the virtual environment by adding it to the codebase or the script of the virtual environment
6	Test and refine the integration	Test and refine the integration by conducting usability testing and user acceptance testing to ensure that the integration works as intended and is user-friendly
7	Iterate and improve the integration	Iterate and improve the integration based on user feedback and observations. This may involve refining the ChatGPT model, improving the virtual environment, or enhancing the user interface

References

1. Adarkwah, M. A., Amponsah, S., Van Wyk, M. M., Huang, R., Tlili, A., Shehata, B., Metwally, A. H. S., & Wang, H. (2023a). Awareness and acceptance of ChatGPT as a generative conversational AI for transforming education by Ghanaian academics: A two-phase study. *Journal of Applied Learning and Teaching*, 6(2), Article 2. <https://doi.org/10.37074/jalt.2023.6.2.26>
2. Adarkwah, M. A., Ying, C., Mustafa, M. Y., & Huang, R. (2023b). Prediction of learner information-seeking behavior and classroom engagement in the advent of ChatGPT. In C. Anutariya, D. Liu, Kinshuk, A. Tlili, J. Yang, & M. Chang (Eds.), *Smart Learning for a Sustainable Society* (pp. 117–126). Springer Nature. https://doi.org/10.1007/978-981-99-5961-7_13
3. Akour, I. A., Al-Maroof, R. S., Alfaisal, R., & Salloum, S. A. (2022). A conceptual framework for determining metaverse adoption in higher institutions of gulf area: An empirical study using hybrid SEM-ANN approach. *Computers and Education: Artificial Intelligence*, 3, 100052. <https://doi.org/10.1016/j.caiei.2022.100052>
4. Almalki, M., & Azeez, F. (2020). Health Chatbots for Fighting COVID-19: A Scoping Review. *Acta Informatica Medica: AIM: Journal of the Society for Medical Informatics of Bosnia &*

- Herzegovina: Casopis Drustva Za Medicinsku Informatiku BiH*, 28(4), 241–247. <https://doi.org/10.5455/aim.2020.28.241-247>
5. Antaki, F., Touma, S., Milad, D., El-Khoury, J., & Duval, R. (2023). *Evaluating the performance of ChatGPT in ophthalmology: An analysis of its successes and shortcomings* (p. 2023.01.22.23284882). medRxiv <https://doi.org/10.1101/2023.01.22.23284882>
 6. Bojic, L. (2022). Metaverse through the prism of power and addiction: What will happen when the virtual world becomes more attractive than reality? *European Journal of Futures Research*, 10(1), 22. <https://doi.org/10.1186/s40309-022-00208-4>
 7. Brandtzaeg, P. B., Skjuve, M., & Følstad, A. (2022). My AI friend: How users of a social chatbot understand their human–AI friendship. *Human Communication Research*, 48(3), 404–429. <https://doi.org/10.1093/hcr/hqac008>
 8. Caldarini, G., Jaf, S., & McGarry, K. (2022). A literature survey of recent advances in chatbots. *Information*, 13(1), Article 1. <https://doi.org/10.3390/info13010041>
 9. Cheng, R., Wu, N., Chen, S., & Han, B. (2022). Will metaverse be NextG internet? Vision, hype, and reality. *IEEE Network*, 36(5), 197–204. <https://doi.org/10.1109/MNET.117.2200055>
 10. De Cicco, R., Silva, S. C., & Alparone, F. R. (2020). Millennials' attitude toward chatbots: An experimental study in a social relationship perspective. *International Journal of Retail & Distribution Management*, 48(11), 1213–1233. <https://doi.org/10.1108/IJRDM-12-2019-0406>
 11. Cotton, D. R. E., Cotton, P. A., & Shipway, J. R. (2023). Chatting and cheating: Ensuring academic integrity in the era of ChatGPT. *Innovations in Education and Teaching International*, 1–12. <https://doi.org/10.1080/14703297.2023.2190148>
 12. Cox, C., & Tzoc, E. (2023). ChatGPT: Implications for academic libraries. *College & Research Libraries News*. <https://doi.org/10.5860/crln.84.3.99>
 13. Dowling, M., & Lucey, B. (2023). ChatGPT for (finance) research: The Bananarama conjecture. *Finance Research Letters*, 103662. <https://doi.org/10.1016/j.frl.2023.103662>
 14. Durall, E., & Kapros, E. (2020). Co-design for a competency self-assessment chatbot and survey in science education. In P. Zaphiris & A. Ioannou (Eds.), *Learning and collaboration technologies. Human and technology ecosystems* (pp. 13–24). Springer International Publishing. https://doi.org/10.1007/978-3-030-50506-6_2
 15. Dwivedi, Y. K., Hughes, L., Baabdullah, A. M., Ribeiro-Navarrete, S., Giannakis, M., Al-Debei, M. M., Dennehy, D., Metri, B., Buhalis, D., Cheung, C. M. K., Conboy, K., Doyle, R., Dubey, R., Dutot, V., Felix, R., Goyal, D. P., Gustafsson, A., Hinsch, C., Jebabli, I., ... Wamba, S. F. (2022). Metaverse beyond the hype: Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. *International Journal of Information Management*, 66, 102542. <https://doi.org/10.1016/j.ijinfomgt.2022.102542>
 16. Dwivedi, Y. K., Hughes, L., Ismagilova, E., Aarts, G., Coombs, C., Crick, T., Duan, Y., Dwivedi, R., Edwards, J., Eirug, A., Galanos, V., Ilavarasan, P. V., Janssen, M., Jones, P., Kar, A. K., Kizgin, H., Kronemann, B., Lal, B., Lucini, B., ... Williams, M. D. (2021). Artificial Intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. *International Journal of Information Management*, 57, 101994. <https://doi.org/10.1016/j.ijinfomgt.2019.08.002>
 17. Dwivedi, Y. K., Hughes, L., Wang, Y., Alalwan, A. A., Ahn, S. J. (Grace), Balakrishnan, J., Barta, S., Belk, R., Buhalis, D., Dutot, V., Felix, R., Filieri, R., Flavián, C., Gustafsson, A., Hinsch, C., Hollensen, S., Jain, V., Kim, J., Krishen, A. S., ... Wirtz, J. (2023). Metaverse marketing: How the metaverse will shape the future of consumer research and practice. *Psychology & Marketing*, 40(4), 750–776. <https://doi.org/10.1002/mar.21767>
 18. Europol. (2023). *ChatGPT (The impact of large language models on law enforcement)*. <https://www.europol.europa.eu/cms/sites/default/files/documents/Tech%20Watch%20Flash%20-%20The%20Impact%20of%20Large%20Language%20Models%20on%20Law%20Enforcement.pdf>
 19. Garavand, A., & Aslani, N. (2022). Metaverse phenomenon and its impact on health: A scoping review. *Informatics in Medicine Unlocked*, 32, 101029. <https://doi.org/10.1016/j.imu.2022.101029>

20. Godwin-Jones, R. (2023). Emerging spaces for language learning: AI bots, ambient intelligence, and the metaverse. *Language Learning & Technology*, 27(2). <https://www.lltjournal.org/item/10125-73501/>
21. Goh, Y. C., Cai, X. Q., Theseira, W., Ko, G., & Khor, K. A. (2020). Evaluating human versus machine learning performance in classifying research abstracts. *Scientometrics*, 125(2), 1197–1212. <https://doi.org/10.1007/s11192-020-03614-2>
22. Guo, H., & Gao, W. (2022). Metaverse-powered experiential situational English-teaching design: An emotion-based analysis method. *Frontiers in Psychology*, 13. <https://www.frontiersin.org/articles/10.3389/fpsyg.2022.859159>
23. Gursoy, D., Malodia, S., & Dhir, A. (2022). The metaverse in the hospitality and tourism industry: An overview of current trends and future research directions. *Journal of Hospitality Marketing & Management*, 31(5), 527–534. <https://doi.org/10.1080/19368623.2022.2072504>
24. Guzman, A. L., & Lewis, S. C. (2020). Artificial intelligence and communication: A human-machine communication research agenda. *New Media & Society*, 22(1), 70–86. <https://doi.org/10.1177/1461444819858691>
25. Haesvoets, T., De Cremer, D., Dierckx, K., & Van Hiel, A. (2021). Human-machine collaboration in managerial decision making. *Computers in Human Behavior*, 119, 106730. <https://doi.org/10.1016/j.chb.2021.106730>
26. Hoffman, M., Kahn, L. B., & Li, D. (2018). Discretion in hiring. *The Quarterly Journal of Economics*, 133(2), 765–800. <https://doi.org/10.1093/qje/qjx042>
27. Hwang, G.-J., & Chien, S.-Y. (2022). Definition, roles, and potential research issues of the metaverse in education: An artificial intelligence perspective. *Computers and Education: Artificial Intelligence*, 3, 100082. <https://doi.org/10.1016/j.caai.2022.100082>
28. Io, H. N., & Lee, C. B. (2017). Chatbots and conversational agents: A bibliometric analysis. In *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (pp. 215–219). <https://doi.org/10.1109/IEEM.2017.8289883>
29. Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günemann, S., Hüllermeier, E., Krusche, S., Kutyniok, G., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., ... Kasneci, G. (2023). *ChatGPT for good? On opportunities and challenges of large language models for education*. EdArXiv <https://doi.org/10.35542/osf.io/5er8f>
30. Kirkpatrick, K. (2022). Applying the metaverse. *Communications of the ACM*, 65(11), 16–18. <https://doi.org/10.1145/3565470>
31. Korzynski, P., Mazurek, G., Altmann, A., Ejdyns, J., Kazlauskaitė, R., Paliszkiewicz, J., Wach, K., & Ziembka, E. (2023). Generative artificial intelligence as a new context for management theories: Analysis of ChatGPT. *Central European Management Journal* (ahead-of-print). <https://doi.org/10.1108/CEMJ-02-2023-0091>
32. Kumar, J. A. (2021). Educational chatbots for project-based learning: Investigating learning outcomes for a team-based design course. *International Journal of Educational Technology in Higher Education*, 18(1), 65. <https://doi.org/10.1186/s41239-021-00302-w>
33. Kung, T. H., Cheatham, M., Medenilla, A., Sillos, C., Leon, L. D., Elepaño, C., Madriaga, M., Aggabao, R., Diaz-Candido, G., Maningo, J., & Tseng, V. (2023). Performance of ChatGPT on USMLE: Potential for AI-assisted medical education using large language models. *PLOS Digital Health*, 2(2), e0000198. <https://doi.org/10.1371/journal.pdig.0000198>
34. Kye, B., Han, N., Kim, E., Park, Y., & Jo, S. (2021). Educational applications of metaverse: Possibilities and limitations. *Journal of Educational Evaluation for Health Professions*, 18, 32. <https://doi.org/10.3352/jeehp.2021.18.32>
35. Lim, W. Y. B., Xiong, Z., Niyato, D., Cao, X., Miao, C., Sun, S., & Yang, Q. (2022). Realizing the metaverse with edge intelligence: A match made in heaven. *IEEE Wireless Communications*, 1–9. <https://doi.org/10.1109/MWC.018.2100716>
36. Lin, Z. (2023). *Why and how to embrace AI such as ChatGPT in your academic life*. PsyArXiv <https://doi.org/10.31234/osf.io/sdx3j>
37. Lippert, A., Shubbeck, K., Morgan, B., Hampton, A., & Graesser, A. (2020). Multiple agent designs in conversational intelligent tutoring systems. *Technology, Knowledge and Learning*, 25(3), 443–463. <https://doi.org/10.1007/s10758-019-09431-8>

38. Lund, B. D., & Wang, T. (2023). Chatting about ChatGPT: How may AI and GPT impact academia and libraries? *Library Hi Tech News* (ahead-of-print). <https://doi.org/10.1108/LHTN-01-2023-0009>
39. Matsubara, M., & Oguchi, M. (2010). Evaluation of metaverse server in a widely-distributed environment. In R. Meersman, T. Dillon, & P. Herrero (Eds.), *On the move to meaningful internet systems: OTM 2010 workshops* (pp. 307–316). Springer. https://doi.org/10.1007/978-3-642-16961-8_49
40. McAdoo, T. (2023). *How to cite ChatGPT*. <https://apastyle.apa.org/> <https://apastyle.apa.org/blog/how-to-cite-chatgpt>
41. Ng, D. T. K. (2022). What is the metaverse? Definitions, technologies and the community of inquiry. *Australasian Journal of Educational Technology*, 38(4), Article 4. <https://doi.org/10.14742/ajet.7945>
42. Njoku, J. N., Nwakanma, C. I., Amaizu, G. C., & Kim, D.-S. (2023). Prospects and challenges of metaverse application in data-driven intelligent transportation systems. *IET Intelligent Transport Systems*, 17(1), 1–21. <https://doi.org/10.1049/itr2.12252>
43. Panda, S., & Kaur, N. (2023). Exploring the viability of ChatGPT as an alternative to traditional chatbot systems in library and information centers. *Library Hi Tech News* (ahead-of-print). <https://doi.org/10.1108/LHTN-02-2023-0032>
44. Park, S.-M., & Kim, Y.-G. (2022). A metaverse: Taxonomy, components, applications, and open challenges. *IEEE Access*, 10, 4209–4251. <https://doi.org/10.1109/ACCESS.2021.3140175>
45. Radziwill, N. M., & Benton, M. C. (2017). *Evaluating quality of chatbots and intelligent conversational agents* (arXiv:1704.04579). arXiv <https://doi.org/10.48550/arXiv.1704.04579>
46. Ramesh, A., & Chawla, V. (2022). Chatbots in marketing: A literature review using morphological and co-occurrence analyses. *Journal of Interactive Marketing*, 57(3), 472–496. <https://doi.org/10.1177/10949968221095549>
47. Rao, H., Leung, C., & Miao, C. (2023). *Can ChatGPT assess human personalities? A general Evaluation framework* (arXiv:2303.01248). arXiv <https://doi.org/10.48550/arXiv.2303.01248>
48. Sun, Y., Xu, Y., Cheng, C., Li, Y., Lee, C. H., & Asadipour, A. (2022). Travel with Wander in the Metaverse: An AI chatbot to Visit the Future Earth. In *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)* (pp. 1–6). <https://doi.org/10.1109/MMSP55362.2022.9950031>
49. Sá, M. J., & Serpa, S. (2023). Metaverse as a learning environment: Some considerations. *Sustainability*, 15(3), Article 3. <https://doi.org/10.3390/su15032186>
50. Tellols, D., Lopez-Sanchez, M., Rodríguez, I., Almajano, P., & Puig, A. (2020). Enhancing sentient embodied conversational agents with machine learning. *Pattern Recognition Letters*, 129, 317–323. <https://doi.org/10.1016/j.patrec.2019.11.035>
51. Tlili, A., Shehata, B., Adarkwah, M. A., Bozkurt, A., Hickey, D. T., Huang, R., & Agyemang, B. (2023). What if the devil is my guardian angel: ChatGPT as a case study of using chatbots in education. *Smart Learning Environments*, 10(1), 15. <https://doi.org/10.1186/s40561-023-00237-x>
52. Tlili, A., Huang, R., Shehata, B., Liu, D., Zhao, J., Metwally, A. H. S., Wang, H., Denden, M., Bozkurt, A., Lee, L.-H., Beyoglu, D., Altinay, F., Sharma, R. C., Altinay, Z., Li, Z., Liu, J., Ahmad, F., Hu, Y., Salha, S., ... Burgos, D. (2022). Is Metaverse in education a blessing or a curse: A combined content and bibliometric analysis. *Smart Learning Environments*, 9(1), 24. <https://doi.org/10.1186/s40561-022-00205-x>
53. Van Wyk, M. M., Adarkwah, M., & Amponsah, S. (2023). Why all the hype about ChatGPT? Academics' views of a chat-based conversational learning strategy at an open distance e-learning institution. *Open Praxis*, 15, 214–225. <https://doi.org/10.55982/openpraxis.15.3.563>
54. Wang, T., Li, J., Kong, Z., Liu, X., Snoussi, H., & Lv, H. (2021). Digital twin improved via visual question answering for vision-language interactive mode in human-machine collaboration. *Journal of Manufacturing Systems*, 58, 261–269. <https://doi.org/10.1016/j.jmsy.2020.07.011>
55. Wang, Y., Su, Z., Zhang, N., Xing, R., Liu, D., Luan, T. H., & Shen, X. (2023). A survey on metaverse: Fundamentals, security, and privacy. *IEEE Communications Surveys & Tutorials*, 25(1), 319–352. <https://doi.org/10.1109/COMST.2022.3202047>

56. Ward, W., Cole, R., Bolaños, D., Buchenroth-Martin, C., Svirsky, E., & Weston, T. (2013). My science tutor: A conversational multimedia virtual tutor. *Journal of Educational Psychology, 105*, 1115–1125. <https://doi.org/10.1037/a0031589>
57. Wollny, S., Schneider, J., Di Mitri, D., Weidlich, J., Rittberger, M., & Drachsler, H. (2021). Are we there yet?—A Systematic literature review on chatbots in education. *Frontiers in Artificial Intelligence, 4*, 654924. <https://doi.org/10.3389/frai.2021.654924>
58. Wong, D., & Floridi, L. (2022). Meta's oversight board: A review and critical assessment. *Minds and Machines*. <https://doi.org/10.1007/s11023-022-09613-x>
59. Xiao, Z., Zhou, M. X., Liao, Q. V., Mark, G., Chi, C., Chen, W., & Yang, H. (2020). Tell me about yourself: Using an AI-powered chatbot to conduct conversational surveys with open-ended questions. *ACM Transactions on Computer-Human Interaction, 27*(3), 15:1–15:37. <https://doi.org/10.1145/3381804>
60. Xu, M., Ng, W. C., Lim, W. Y. B., Kang, J., Xiong, Z., Niyato, D., Yang, Q., Shen, X., & Miao, C. (2023). A full dive into realizing the edge-enabled metaverse: Visions, enabling technologies, and challenges. *IEEE Communications Surveys & Tutorials, 25*(1), 656–700. <https://doi.org/10.1109/COMST.2022.3221119>
61. Yilma, B. A., Panetto, H., & Naudet, Y. (2021). Systemic formalisation of cyber-physical-social system (CPSS): A systematic literature review. *Computers in Industry, 129*, 103458. <https://doi.org/10.1016/j.compind.2021.103458>
62. Zeng, J., Yang, L. T., Lin, M., Ning, H., & Ma, J. (2020). A survey: Cyber-physical-social systems and their system-level design methodology. *Future Generation Computer Systems, 105*, 1028–1042. <https://doi.org/10.1016/j.future.2016.06.034>
63. Zhai, X. (2022). *ChatGPT user experience: Implications for education* (SSRN Scholarly Paper No. 4312418). <https://doi.org/10.2139/ssrn.4312418>
64. Zhang, J. J., Wang, F.-Y., Wang, X., Xiong, G., Zhu, F., Lv, Y., Hou, J., Han, S., Yuan, Y., Lu, Q., & Lee, Y. (2018). Cyber-physical-social systems: The state of the art and perspectives. *IEEE Transactions on Computational Social Systems, 5*(3), 829–840. <https://doi.org/10.1109/TCSS.2018.2861224>
65. Zhang, X., Chen, Y., Hu, L., & Wang, Y. (2022). The metaverse in education: Definition, framework, features, potential applications, challenges, and future research topics. *Frontiers in Psychology, 13*. <https://www.frontiersin.org/articles/10.3389/fpsyg.2022.1016300>
66. Zhou, Y., Yu, F. R., Chen, J., & Kuo, Y. (2020). Cyber-physical-social systems: A state-of-the-art survey, challenges and opportunities. *IEEE Communications Surveys & Tutorials, 22*(1), 389–425. <https://doi.org/10.1109/COMST.2019.2959013>

Generating Artistic Portrait Drawings from Images



Ran Yi, Yong-Jin Liu, Yu-Kun Lai, and Paul L. Rosin

Abstract This chapter addresses generating artistic portrait drawings (APDrawings) from images, and we focus on two methods based on generative adversarial networks (GANs). We first introduce the genre of portrait line drawings, and review some existing methods for generating them from images. We also describe the Artistic Portrait Drawing (APDrawing) dataset, which contains 140 high-resolution face photos and corresponding portrait drawings executed by a professional artist. We then describe the APDrawingGAN method, which is a hierarchical GAN model that learns from paired data of face photos and portrait drawings, and the QMUPD method, which can learn from unpaired data of face photos and drawing. APDrawingGAN uses a novel distance transform loss to learn stroke lines in the drawings, and a local transfer loss to capture different drawing styles for different facial regions. QMUPD uses an asymmetric cycle mapping to preserve important facial features, and a quality metric to guide the generation towards high-quality drawings. We further introduce some recent developments which are based on multiple scale analysis, 3D information and multi-modal information. Finally, we describe the evaluation of artistic portrait drawings, which is a challenging task since there are many possible drawings that would be considered by experts to be acceptable.

R. Yi (✉)

Department of Computer Science and Engineering, Shanghai Jiao Tong University,
Shanghai 200240, China
e-mail: ranyi@sjtu.edu.cn

Y.-J. Liu

MOE-Key Laboratory of Pervasive Computing, BNRIst, Department of Computer Science
and Technology, Tsinghua University, Beijing 100084, China
e-mail: liuyongjin@tsinghua.edu.cn

Y.-K. Lai · P. L. Rosin

School of Computer Science and Informatics, Cardiff University, CF24 3AA Cardiff, UK
e-mail: laiy4@cardiff.ac.uk

P. L. Rosin

e-mail: rosinpl@cardiff.ac.uk

1 Introduction

This chapter focuses on the genre of portrait line drawings, and is therefore circumscribed both by medium (typically pen or pencil) and topic (typically human faces, although some artists specialise in non-humans, e.g. the horse portraits painted by George Stubbs). Nevertheless, portrait line drawings still cover a large range of styles, in part due to their long historical development, as well as their different applications. For instance, more than 2000 years ago, the ancient Greeks produced thousands of painted pottery vases, and Fig. 1a shows a portrait line drawing from the red-figure classical period, which looks fairly contemporary. While these drawings consisted of clean outlines, the consequence of eliminating colour and texture and using just lines, is that it becomes hard to capture shading. Removing colour also introduces problems, although both artists and researchers have developed solutions [2]. One method for retaining some aspect of shading is to introduce hatching Fig. 1b, although the main methods described in this chapter aim towards generating drawings with fewer lines. This is in the spirit of our earlier work whose goal was to perform minimal rendering with lines as well as regions [29]. Depending on their goals, artists might switch between the different styles, as seen in the example by Leonardo da Vinci in Fig. 1c. Here, one version of the head is drawn in detail with careful hatching to provide good modelling of the surface, whereas another version is more an outline for quickly exploring some possible design options. Moving forwards to the twentieth century, Fig. 2 shows different styles, e.g. minimal/clean, messy, highly stylised, and minimal/cartoon. Figure 3 shows how, from the basis of similar photographs, an artist can derive dissimilar artworks, e.g. realist versus highly stylised.

Moving to computer generated line portraits, early work in the non-photorealistic rendering (NPR) community developed various approaches, many of which involved lines and solid black regions. For instance, Gooch et al. [12] extracted lines using difference of Gaussian filters (DoG) at multiple scales followed by global threshold-



Fig. 1 Examples of portrait line drawings from the last 2500 years. **a** section of a vase in the Attic red-figure style by Aristophanes (410–400 B.C.), **b** Self-portrait with Long Bushy Hair (1629–1633) by Rembrandt, **c** Study for the head of Leda (1503–1507) by Leonardo da Vinci

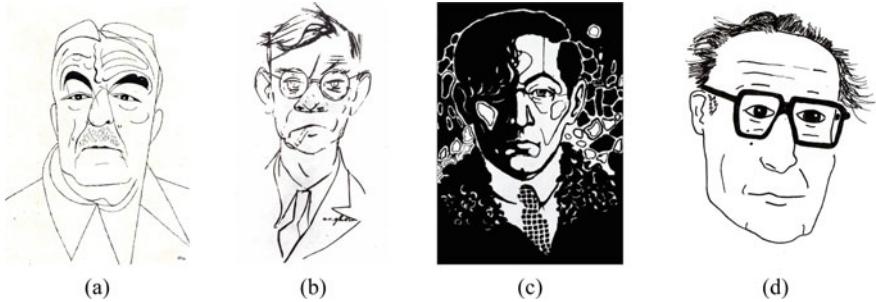


Fig. 2 Twentieth century portrait line drawings. **a** Fernand Léger (1956) by Adolf Hoffmeister, **b** Hans Fallada (1943) by Erich Ohser, **c** Self portrait (1921) by Geo Milev, **d** André Gorz (2022) by HerB104

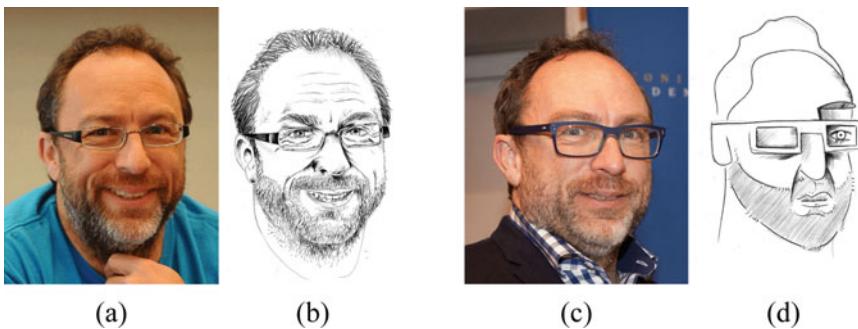


Fig. 3 Portraits of Jimmy Wales by Jericó Delayah derived from the source photographs

ing. The lines were combined with dark regions that were extracted from the source intensity image by thresholding. As Fig. 4b shows the results are reasonable, but in the absence of additional filtering are somewhat noisy even for simple input images. The results from Meng et al. [24] in Fig. 4c are meant to simulate paper-cuts. These are effectively binary renderings with the extra constraint that the black pixels form a single connected region. Their approach is more complex, involving a hierarchical composition model which represents the face by an AND-OR graph in which the nodes represent facial components. Facial features are located in the source image by fitting an active appearance model [9], from which local thresholding produces a set of “proposal” regions which are matched to the graph. Finally, post-processing is applied to extract the hair and clothing using graph cut segmentation, and enforce connectivity by inserting some curves. Like Gooch et al., Rosin and Lai [29] combine lines and regions. In their case lines are extracted using Kang et al.’s [19] coherent line drawing algorithm, which constructs a smooth edge tangent flow following the salient image edges that determines the kernel shape for the DoG filter. Both black and white (negative) lines are extracted. Regions are extracted by applying thresholding followed by GrabCut [33]. The results shown in Fig. 4d are cleaner than those of Gooch et al., although some details have been lost (e.g. the jaw-line).

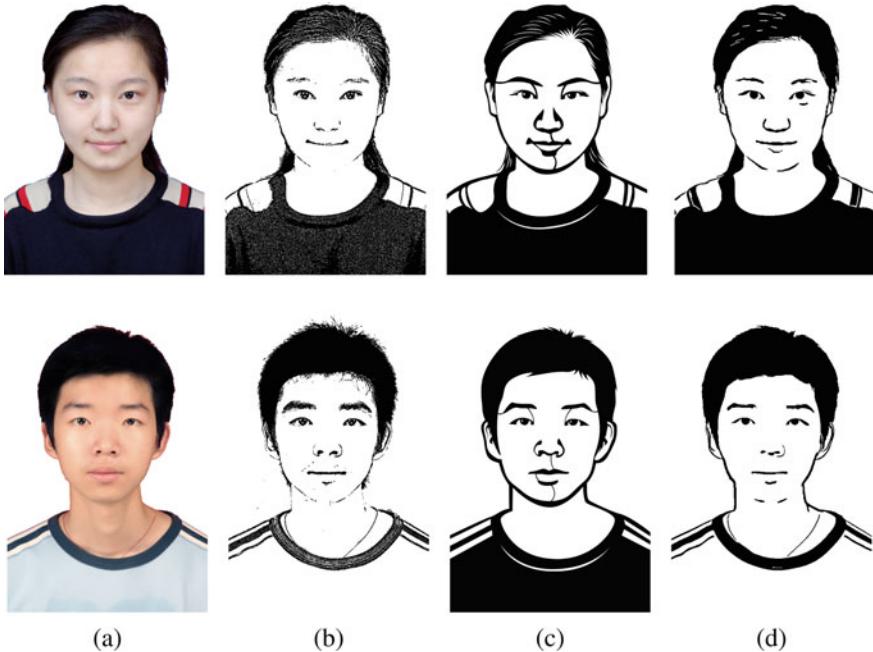


Fig. 4 Various black and white line and region NPR portrait renderings. **a** Source image, **b** Gooch et al. [12], **c** Meng et al. [24], **d** Rosin and Lai [29]

Figure 5 shows further examples of line drawings, some specifically designed for portraits, while others are general purpose such as Chiu et al.’s circular scribble art [6]. This produces a whimsical circular scribble pattern that is attractive, but does not really capture the portrait’s identity, see Fig. 5b, c. Their system first generates a virtual tracing path that takes the image’s intensity and edge structure into account. Circular scribbles are synthesized along the virtual path, with the circle radius controlled by the local intensity. Figure 5d shows a binary version of the heavily stylised Julian Opie effect produced by Rosin and Lai’s portrait stylisation method [30]; it uses the black and white lines and regions produced by their earlier minimal rendering style [29] and a template to create the facial features. Stippling is a popular general purpose image stylisation approach, widely used both by artists and the wider community. Given that it uses a huge number of graphical elements (stipples), it is the antithesis of this chapter’s focus on more minimal stylisation. However, there is a portrait-specific variant of stippling called *hedcut*, shown in Fig. 5e that produces a clearer effect. This result was generated with Son et al.’s [38] algorithm, that uses a regularly spaced grid of dots and hatching lines which are deformed to fit the image. Rosin and Lai [31] created an engraving stylisation using a dither matrix (i.e. a spatially-varying threshold) that generates a pattern of black and white lines forming cross hatching. A simple cylindrical model of the face warped the dither matrix so that the lines curve around the face, providing a pseudo-3D effect—see

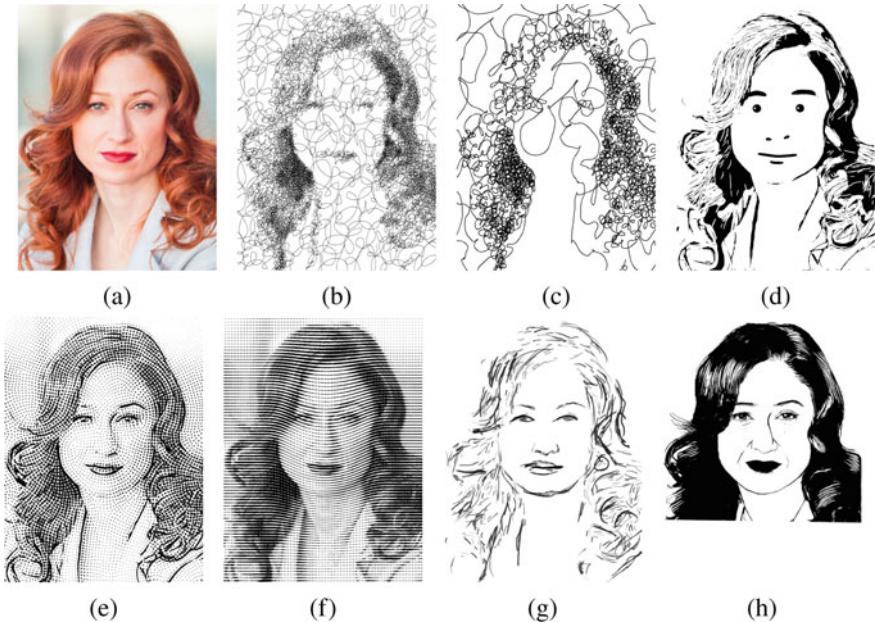


Fig. 5 Various black and white portrait stylisations. **a** original image, **b, c** Chiu et al.’s circular scribble art [6], **d** Rosin and Lai’s ‘Julian Opie’ style [30], **e** Son et al.’s hedcut [38], **f** Rosin and Lai’s engraving [31], **g** Berger et al.’s [3] portrait sketching, **h** Yi et al.’s APDrawingGAN [49]

Fig. 5f. Berger et al. [3] use the statistics of a set of drawings of artists to drive an algorithm that creates a contour image, detects facial features, and then modifies the face geometry to follow the specific artist’s geometric style. Finally, contours are drawn using strokes from the artist’s stroke database, see Fig. 5g. The style is intentionally sketchy, which enables it to effectively hide errors in rendering. The last result, Fig. 5h, shows a stylisation by APDrawingGAN [49]—this method will be described in more detail in the next section.

It can be seen that generating high quality portrait line drawings is challenging, and this comes from two fronts. First, the use of a sparse set of lines rather than a dense set of graphical primitives (e.g. painting strokes or stippling) means that any errors in these lines is significant. A mislocalisation or deformation of even a single line can become evident to the viewer, and spoil the artistic effect. In comparison, an error in an individual stipple will barely be visible. Second, the human visual system is especially sensitive to the human face, and will quickly perceive any errors. For instance, a missing eye on a portrait is unacceptable, and of much greater consequence than, e.g. a missing finger.

2 APDrawingGAN

APDrawingGAN [49, 50] is a Hierarchical Generative Adversarial Network (GAN) model dedicated to face structure and Artistic Portrait Line Drawing (APDrawing) styles for transforming face photos to high-quality APDrawings. To effectively learn different drawing styles for different facial regions, the APDrawingGAN architecture involves several local networks dedicated to facial feature regions, along with a global network to capture holistic characteristics. To further cope with line-stroke-based style and imprecisely located elements in artists' drawings, APDrawingGAN proposed a novel distance transform (DT) loss to learn stroke lines in APDrawings.

2.1 Challenges

APDrawingGAN addressed the following five challenges to improve the quality of artistic portrait drawings (APDrawing). In addition to the two previously mentioned, namely sparse graphical elements and sensitivity of the human visual system to faces, some additional challenges include:

- In previous methods, different facial areas may be rendered in different styles (*e.g.*, eyes vs. hair).
- APDrawings will make some trade-offs to the elements of the original face, posing a challenge for methods based on pixel correspondence (*e.g.*, Pix2Pix [18]).
- In APDrawings, some lines are not directly related to low level features in the view or photograph of the person.

Figure 6 gives some examples. These examples include lines in the hair indicating the flow, or lines indicating the presence of facial features even if the image contains no intensity or colour discontinuities. Such elements of the drawings are hard to learn. Therefore, many image style transfer algorithms (*e.g.*, [11, 18, 20, 21, 37, 57]) inevitably fail to produce good and expressive artistic portraits (Fig. 7).

To solve the above challenges, APDrawingGAN firstly uses a Hierarchical GAN architecture for artistic portrait drawing synthesis from a face photo, which can generate high-quality and expressive artistic portrait drawings. To best emulate artists, who use multiple graphical elements when creating a drawing, APDrawingGAN separates the GAN's rendered output into multiple layers, each of which is controlled by separate loss functions. The APDrawing dataset is constructed to facilitate research in this area, and contains 140 high-resolution face photos and corresponding portrait drawings executed by a professional artist. Fig. 7 shows the qualitative results of APDrawingGAN and the comparison with seven neural style transfer and image-to-image translation methods.



Fig. 6 Some examples of image pairs (each pair contains a face photo and an artist’s portrait drawing) in the APDrawing dataset [49]

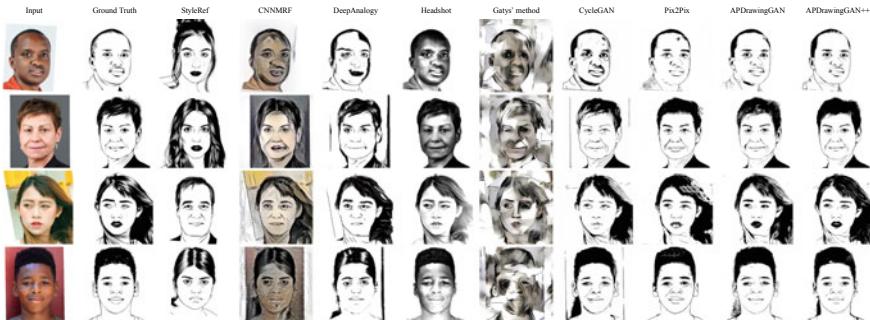


Fig. 7 Qualitative results of APDrawingGAN/APDrawingGAN++ and comparison with seven neural style transfer and image-to-image translation methods. From left to right: input face photos, ground truth APDrawings, the randomly-chosen style images for methods which take one content and one style image as input, CNNMRF [20] results, Deep Image Analogy [21] results, Headshot Portrait [37] results, Gatys [11] results, CycleGAN [57] results, Pix2Pix [18] results, the original APDrawingGAN [49] results, APDrawingGAN++ [47] results. Compared with the original APDrawingGAN, APDrawingGAN++ uses auto-encoders, classifiers for lip and hair, and line continuity loss for better qualitative results

2.2 Technical Details of APDrawingGAN

The process of learning to transform face photos to APDrawings can be modeled as a function Ψ which maps the face photo domain \mathcal{P} into a black-and-white line-stroke-based APDrawing domain \mathcal{A} . The function Ψ is learnt from paired training data $S_{data} = \{(p_i, a_i) | p_i \in \mathcal{P}, a_i \in \mathcal{A}, i = 1, 2, \dots, N\}$, where N is the number of photo-APDrawing pairs in the training set. The discussion in this section focuses on the extended version called APDrawingGAN++ [47], which uses additional auto-encoders for fine facial features, classification for lips and hair, and line continuity

loss to improve the line quality. To streamline the text, unless explicitly specified, we use APDrawingGAN to refer to the extended model.

APDrawingGAN consists of a generator G and a discriminator D , both of which are convolutional neural networks specifically designed for line drawing-based APDrawings in the style of artist's drawings. The generator G learns the APDrawing of the output \mathcal{A} , while the discriminator D serves to determine whether an image is real or generated by the generator.

The discriminator D is trained to classify the real $a_i \in \mathcal{A}$ and the synthetic image $G(p_i)$, $p_i \in \mathcal{P}$ as accurately as possible, while G is trained to minimize this probability. The loss function, denoted $L(G, D)$, is specifically designed with five terms $L_{adv}(G, D)$, $L_{\mathcal{L}_1}(G, D)$, $L_{DT}(G, D)$, $L_{local}(G, D)$ and $L_{cont_i}(G, D)$. Then the function Ψ can be formulated using the function $L(G, D)$ to solve the following min-max problem:

$$\begin{aligned} \min_G \max_D L(G, D) = & L_{adv}(G, D) + \lambda_1 L_{\mathcal{L}_1}(G, D) \\ & + \lambda_2 L_{DT}(G, D) + \lambda_3 L_{local}(G, D) + \lambda_4 L_{cont_i}(G, D). \end{aligned} \quad (1)$$

2.2.1 Hierarchical Generator G

The hierarchical generator G converts the input face photos into APDrawings. The model is trained on one style of APDrawings at a time. In the hierarchy of $G = \{G_{global}, G_{l*}, E_*, C_*, G_{fusion}\}$, G_{global} is a global generator, $G_{l*} = \{G_{l_eye_l}, G_{l_eye_r}, G_{l_nose}, G_{l_mouth}, G_{l_hair}, G_{l_bg}\}$ is a set of six local generators. $E_* = \{E_{eye_l}, E_{eye_r}, E_{nose}, E_{lip_b}, E_{lip_w}\}$ is a set of five auto-encoders, $C_* = \{Clip, Chair\}$ is a set of two classifiers and G_{fusion} is a fusion network.

The generator G uses the U-Net structure [28]. $G_{l_eye_l}$, $G_{l_eye_r}$, G_{l_nose} and G_{l_mouth} are all U-Nets with three downward and three upward convolutions. G_{l_hair} and G_{l_bg} are U-Nets with four downward and four upward convolution blocks. In G_{l*} , the role of the local generator is to learn the drawing styles of different local facial features; for example, the hair style for hair (i.e., capturing the soft, flowing details of individual strands of hair with short or long strokes), the delicate line style for eyes and noses, and the solid or line style for mouths. A U-Net with skip connections can incorporate multi-scale features and provide sufficient but not excessive flexibility to learn the artist's drawing techniques for different facial regions in APDrawings.

Local Generators. The inputs of $G_{l_eye_l}$, $G_{l_eye_r}$, G_{l_nose} , and G_{l_mouth} are local regions centered on facial elements (i.e., left eye, right eye, nose and mouth) as the centered local regions, obtained from the MTCNN model [54]. The input of G_{lbg} is the background region detected by the portrait segmentation method [36]. The input of G_{hair} is the remaining region in the face photo. The outputs of all local generators are blended into an aggregated picture I_{local} by using min pooling in the overlapping regions. This min pooling effectively preserves the responses of individual local generators, because in artistic pictures, low intensities are considered as the responses of black pixels.

Global Generator. G_{global} is a U-Net with eight lower and eight upper convolutional blocks, which handles the global structure of the face. G_{fusion} consists of a vanilla convolution block (the feature map size stays the same), six residual blocks and a final convolution layer. G_{fusion} is used to fuse I_{local} and I_{global} (i.e., the output of G_{global}) to obtain the final synthetic map of G . In many previous GAN models (e.g., [13, 17]), some noise is usually input or injected to the generator network. Following [18], instead of adding noise explicitly in G , APDrawingGAN uses dropout [39] as noise in the U-Net block.

Fusion Network. The fusion Network G_{fusion} is used to fuse the output from the local and global generators together for final portrait drawing synthesis. This block helps combine different drawing techniques learnt by different generators (G_{global} , G_{l*}).

Handling Multiple Styles for Lips and Eyes. In the APDrawing dataset, lips and hair exhibit multiple styles, e.g., white/black lips, and dark/middle/light hair (Fig. 6). We use two classifiers for lip and hair (C_{lip} , C_{hair}) to detect the target style for the lip and hair regions respectively, and the detected class information is then used to guide the generation toward the desired style.

Autoencoders for Fine APDrawing. In the original APDrawingGAN [49], the generator G only consists of local generators, a global generator and a fusion network, where the main loss function was calculated on the fused result output from the fusion network, while the local generators' outputs are only supervised by a local loss. Therefore, the local drawings output from the local generators $G_{l_eye_l}$, $G_{l_eye_r}$, G_{l_nose} , G_{l_lip} are not as delicate as the artist drawn drawings. In APDrawingGAN++ [47], a set of auto-encoders E_{eye_l} , E_{eye_r} , E_{nose} , $E_{lip_b/w}$ (corresponding to the left eye, right eye, nose and lip) are designed to improve local drawings and generate better facial feature drawings in fine detail. Both the coarse input and fine output of these auto-encoders are parts of APDrawings. Trained with the APDrawing dataset, each auto-encoder learns a good feature representation and reconstructs high-quality APDrawings close to the artist drawings.

2.2.2 Hierarchical Discriminator D

The discriminator D distinguishes whether the input drawing is a genuine portrait of the artist. In the hierarchy of $D = \{D_{global}, D_{l*}\}$, D_{global} is a global discriminator, and $D_{l*} = \{D_{l_eye_l}, D_{l_eye_r}, D_{l_nose}, D_{l_mouth}, D_{l_hair}, D_{l_bg}\}$ is a set of six local discriminators. D_{global} examines the whole drawing to determine the overall APDrawing features, and the local discriminators in D_{l*} examine different local areas to evaluate the quality of the details.

D_{global} and all local discriminators in D_{l*} use the Markovian discriminator in Pix2Pix [18]. The only difference is the input: the whole drawings or different local regions. The Markovian discriminator processes each 70×70 patch in the input image and examines the style of each patch. Local patches from different granularities (i.e., coarse and fine levels at global and local input) allow the discriminator to

learn local patterns and better discriminate real artists' drawings from synthesized drawings.

2.2.3 Loss Function

There are five terms in the loss function in Eq. 1, which are explained as follows.

Adversarial loss. L_{adv} models the ability of the discriminator to correctly distinguish between true and false APDrawings. According to Pix2Pix [18], the adversarial loss is formulated as

$$L_{adv}(G, D) = \sum_{D_j \in D} \mathbb{E}_{(p_i, a_i) \sim S_{data}} [\log(D_j(p_i, a_i)) + \log(1 - D_j(p_i, G(p_i)))].$$

When $D_j \in D_{l*}$, the images p_i, a_i and $G(p_i)$ are restricted to the local region specified by D_j . Since D maximizes this loss and G minimizes it, L_{adv} forces the synthesized picture to become closer to the target domain \mathcal{A} .

Pixel-wise loss. $L_{\mathcal{L}_1}$ drives the synthesised image close to the ground truth APDrawing image in a pixel-wise way. The loss of $L_{\mathcal{L}_1}$ is computed for each pixel in the entire drawing:

$$L_{\mathcal{L}_1}(G, D) = \mathbb{E}_{(p_i, a_i) \sim S_{data}} [\|G(p_i) - a_i\|_1]. \quad (2)$$

Using \mathcal{L}_1 norm usually results in less blurry output than \mathcal{L}_2 norm, so it is more suitable for APDrawing style.

Line-promoting distance transform loss. Since the position of elements in APDrawings does not precisely correspond to the intensity of the image, L_{DT} is a loss specifically designed to facilitate line strokes in the APDrawing style. L_{DT} is designed to tolerate the small misalignments often found in artist portraits and to better learn the lines in APDrawings. It relies on the Distance Transformation (DT) and Chamfer matching.

A DT (also known as a “distance map”) can be represented as a digital image in which each pixel stores a distance value. Given a real or synthetic APDrawing x , the two DTs of x are defined as the images $I_{DT}(x)$ and $I'_{DT}(x)$: Suppose that \hat{x} is the binarized image of x , each pixel in $I_{DT}(x)$ stores the distance to the closest black pixel in \hat{x} and each pixel in $I'_{DT}(x)$ stores the distance to its nearest white pixel.

Two convolutional neural networks are used to detect the black and white lines in APDrawings, denoted as Θ_b and Θ_w , respectively. The Chamfer matching distance between APDrawings x_1 and x_2 is defined as:

$$\begin{aligned} d_{CM}(x_1, x_2) &= \sum_{(j, k) \in \Theta_b(x_1)} I_{DT}(x_2)(j, k) \\ &\quad + \sum_{(j, k) \in \Theta_w(x_1)} I'_{DT}(x_2)(j, k), \end{aligned} \quad (3)$$

where $I_{DT}(x)(j, k)$ and $I'_{DT}(x)(j, k)$ are the distance values of pixels (j, k) in the images $I_{DT}(x)$ and $I'_{DT}(x)$, respectively. $d_{CM}(x_1, x_2)$ measures the sum of the distances from each line pixel in x_1 to the nearest pixel of the same type (black or white) in x_2 . Then L_{DT} is defined as:

$$L_{DT}(G, D) = \mathbb{E}_{(p_i, a_i) \sim S_{data}} [d_{CM}(a_i, G(p_i)) + d_{CM}(G(p_i), a_i)]. \quad (4)$$

Local transfer loss. L_{local} imposes additional constraints on the intermediate outputs of the six local generators in G_{l*} , which are then used as regularization terms for the loss function. The six local regions of APDrawing x are denoted by $El(x)$, $Er(x)$, $Ns(x)$, $Mt(x)$, $Hr(x)$, and $Bg(x)$. L_{local} is defined as

$$L_{local}(G, D) = \mathbb{E}_{(p_i, a_i) \sim S_{data}} [||G_{l_eye_l}(El(p_i)) - El(a_i)||_1 + ||G_{l_eye_r}(Er(p_i)) - Er(a_i)||_1 + ||G_{l_nose}(Ns(p_i)) - Ns(a_i)||_1 + ||G_{l_mouth}(Mt(p_i)) - Mt(a_i)||_1 + ||G_{l_hair}(Hr(p_i)) - Hr(a_i)||_1 + ||G_{l_bg}(Bg(p_i)) - Bg(a_i)||_1]. \quad (5)$$

Line continuity loss. Line continuity is important in APDrawings, and the lines in the human artist drawings are often continuous. To promote the model to generate more continuous lines, a line continuity loss can be used to guide the model training. A line continuity prediction network R_{conti} is designed to predict the line continuity score from a drawing patch, which is trained from the artist patches (which are assigned the highest continuity score) and manufactured defect patches (which are generated by randomly inverting line or non-line pixels in artist patches and are assigned lower continuity scores).

In detail, the line continuity prediction network R_{conti} , which contains three flat-convolutions and a fully-connected layer, takes an 11×11 patch as input and outputs a single value for line continuity. As described above, the line continuity score of an APDrawing I can be defined as:

$$S_{conti}(x) = \mathbb{E}_{\rho_k \sim P(x)} R_{conti}(\rho_k), x \in I, \quad (6)$$

where $P(x)$ is the set of all patches that are not pure white or pure black, extracted from I , and ρ_k is the k -th patch in this set. The higher the line continuity score, the more continuous the lines in APDrawing I . For face and non-face patches, there are patch sets $P_{face}(x)$ and $P_{non-face}(x)$. And the line continuity loss can be defined as:

$$L_{conti}(G, D) = \mathbb{E}_{(p_i, a_i) \sim S_{data}} \mathbb{E}_{\rho_k \sim P(G(p_i))} w_k (1 - R_{conti}(\rho_k)), \quad (7)$$

where weight $w_k = 2$ if $\rho_k \in P_{face}(G(p_i))$, and $w_k = 1$ if $\rho_k \in P_{nface}(G(p_i))$. Since face patches have a complicated set of lines, the lines in the face area are often less continuous and need to be given higher weights to avoid them being unfairly penalised.

3 Unpaired Portrait Drawing Generation (UPD)

APDrawingGAN [49, 50] introduced in the previous section is trained using paired data consisting of face photos and APDrawings. However, paired data is costly to obtain, requiring professional artists hours to draw each delicate APDrawing. In comparison, unpaired training data collected from websites is easier to obtain. But training APDrawingGAN to perform generation from unpaired training data is more challenging than learning from paired training data, because: (1) Paired training data provides a more direct guidance for learning the photo-to-drawing mapping, while unpaired training data cannot provide such direct guidance; (2) Paired training data is usually specially collected and drawn by a few artists, which means the samples are both high quality and uniform in style, whereas this is less easy to achieve when collecting the necessarily large sets of unpaired samples.

In this section, we introduce the Quality-Metric-guided Unpaired Portrait line Drawing Generation method (QMUPD) [47, 48], which targets the scenario in which only unpaired training data is available. Previous methods for unpaired image-to-image translation [52, 56] use a cycle structure to regularize training. Due to the significant imbalance in information richness between photos and drawings, some existing unpaired transfer methods, such as CycleGAN [56], tend to indiscriminately embed invisible reconstruction information throughout the drawings, resulting in important facial features partially lost in the drawings (*e.g.*, Fig. 8 second column). The problem mentioned above can be solved using a new asymmetric cycle mapping by forcing the reconstruction information to be visible (via truncation loss) and embedded only in selective facial regions (via a relaxed forward cycle consistency loss). Together with local discriminators for eyes, nose and lips, the asymmetric cycle mapping well preserves all important facial features in the generated portraits. By introducing a style classifier and taking into account style features, the Quality-Metric-guided Unpaired Portrait line Drawing Generation can learn to generate multiple styles of portraits using a single network. Figures 8 and 9 show the results of the Quality-Metric-guided Unpaired Portrait line Drawing Generation and comparison methods. Due to the use of unpaired data, we consider three typical APDrawing styles, trained using APDrawing images collected from the internet.

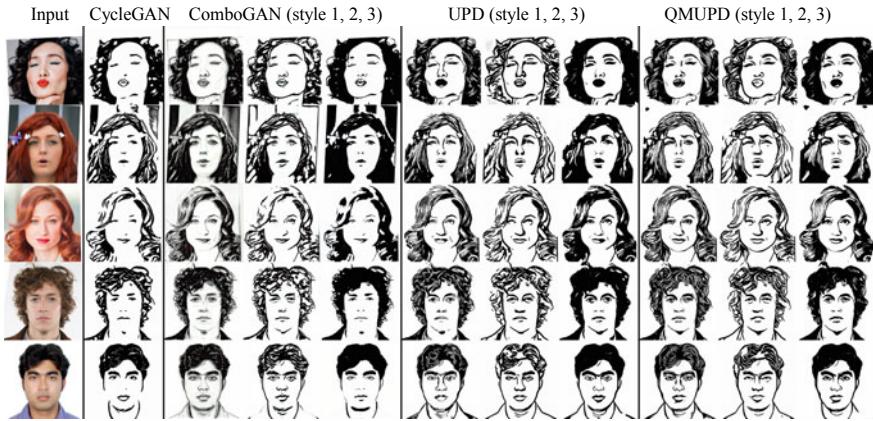


Fig. 8 Quality-Metric-guided Unpaired Portrait line Drawing Generation (QMUPD) qualitative comparisons. From left to right: input face photos, CycleGAN [56] results, ComboGAN [1] results (styles 1, 2, 3), UPD [47] (styles 1, 2, 3), and QMUPD [48] (styles 1, 2, 3). The input face photos are from [32]

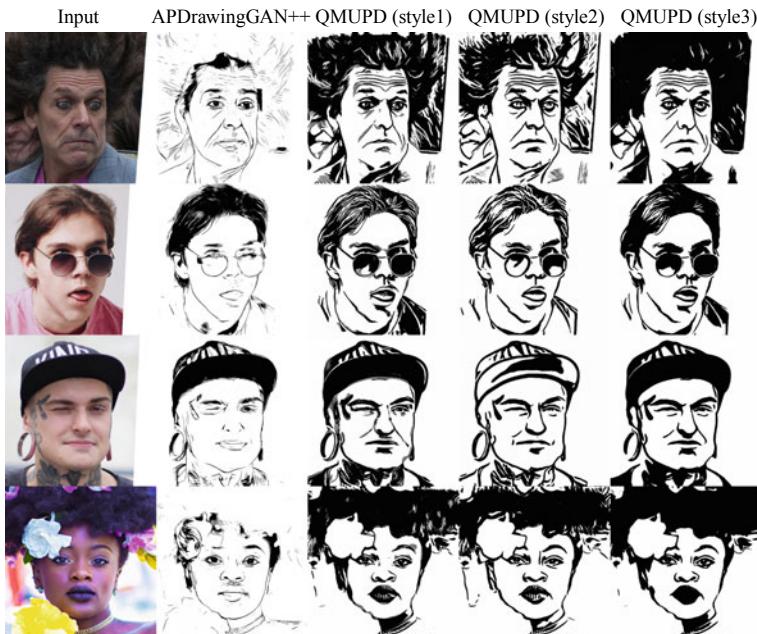


Fig. 9 Quality-Metric-guided Unpaired Portrait line Drawing Generation (QMUPD) qualitative comparisons. From left to right: input face photos, APDrawingGAN++ [50], QMUPD [48] (styles 1, 2, 3). The input face photos are from [32]

3.1 Challenge

In practical applications, the data we have access to are often unpaired. Compared to paired training data, APDrawing generation for learning from unpaired data is more challenging but more relevant. Previous unpaired image-to-image translation methods [52, 56] use a cycle structure to normalize the training. Although cycle consistency loss can be learned from unpaired data, when they are applied to face photo-to-APDrawing translation, due to the apparent imbalance in information richness between the two data types, these methods tend to indiscriminately embed invisible reconstruction information throughout the APDrawing, resulting in degraded quality of the generated APDrawings, such as important facial features are partially lost.

3.2 Quality Metric for APDrawings

For high-quality APDrawing generation, it is not sufficient to decide whether such a drawing is real or fake; the generator needs a quality metric during training for the high-quality synthesized drawing. Observing that humans can easily decide the quality of a portrait line drawing without knowing the original face photo, this section introduces a quality metric for portrait line drawings by learning from human preference, which can then be used to encourage the model to generate good looking portrait line drawings. The metric can be modelled by a regression network to calculate the quality score of each drawing based on human preference and predict the quality score of an APDrawing.

Human preference scores. Human preference scores were obtained based on pairwise comparison between portrait line drawings of the same style. A user study was conducted, where the user was shown three portrait line drawings of the same style in a single question and asked to rank the three drawings. The best of the three shown drawings gets +2 reward score, while the middle placed drawing gets no reward and the worst one gets -2 score. 250 drawings for each of the three target styles were chosen for making the questionnaire and 2450–3450 question responses were collected for each style. After summarizing all question responses for a style, the score for each drawing of this style was calculated and the global ranking was obtained based on the score. Finally, the scores were normalized to the range [0.1, 1] for later steps.

Network architecture. Given the portrait drawing data and the normalized quality score (given by humans), a regression network is trained to predict APDrawing quality. The regression network is based on the Inception v3 [40] architecture. It takes an APDrawing as input and outputs a quality value. Since the quality metric model behaviour is learnt from human evaluation, the predicted score can be used as a constraint item to guide the drawing generator toward better quality.

3.3 Technical Details

The Quality-Metric-guided Unpaired Portrait line Drawing Generation uses a new GAN with an asymmetric cycle structure for face photo to APDrawing conversion without paired training data. Let \mathcal{P} and \mathcal{D} be the face photo domain and the APDrawing domain, and no pairing needs to exist between these two domains. The model uses the training data $S(p) = \{p_i | i = 1, 2, \dots, N\}$ and $S(d) = \{d_j | j = 1, 2, \dots, M\}$ to learn a function Φ that maps from \mathcal{P} to \mathcal{D} . N and M are the numbers of training photos and APDrawings. The asymmetric cyclic mapping model consists of two generators—a generator G that converts face photos to portrait drawings and an inverse generator F that converts drawings back to face photos—and two discriminators, $D_{\mathcal{D}}$ for discriminating generated drawings from real drawings and $D_{\mathcal{P}}$ for discriminating generated photos from real photos.

3.3.1 Face Photo to Drawing Generator G

The generator G takes a face photo p and a style feature s as input and outputs a portrait line graph $G(p, s)$ with a style specified by s .

Style features. A classifier C (based on VGG19) was used to classify the portrait line drawings into three styles, using the network drawing data labelled with style classes. Then, the output of a final fully connected layer and a softmax layer were used to compute a 3-dimensional vector as a style feature for each drawing (including unlabelled ones).

Network structure. G is an encoder-decoder with a residual block [15] in the middle. It starts with a flat convolution (the feature map size stays the same) and two down convolution blocks to encode the face photos and extract useful features. The stylized features are then mapped as 3-channel feature maps and inserted into the network by concatenating with the feature maps of the second lower convolution block. Additional flat convolution is used to merge the style feature maps with the extracted feature maps. After that, the remaining blocks of nine identical structures are used to construct content features and transfer them to the target domain. Then, the output drawing is reconstructed by two upward convolutional blocks and a final convolutional layer.

3.3.2 Drawing Discriminator $D_{\mathcal{D}}$

The drawing discriminator $D_{\mathcal{D}}$ has two tasks: (1) to distinguish the generated portrait line drawings from the real ones, and (2) to classify a drawing into three selected styles, where the real one d is expected to be assigned to the correct style label (given by C) and the generated one $G(p, s)$ is expected to be assigned to the style specified by the 3-dimensional style feature s .

For the first task, to ensure the presence of important facial features in the generated drawings, in addition to the discriminator D that analyzes the whole drawing, three local discriminators D_{ln} , D_{le} , D_{ll} are used here to discriminate the drawing of nose, eyes and lips, respectively. The inputs of these local discriminators are masked drawings where the mask is obtained from the face resolution network [14]. The $D_{\mathcal{D}}$ consists of D , D_{ln} , D_{le} , D_{ll} .

Network structure. The global discriminator D is based on PatchGAN [18] and modified to have two branches. These two branches share three down convolution blocks. The branch D_{rf} includes two down convolution blocks to output the true/false prediction maps for each patch in the drawing. And the other classification branch D_{cls} includes more down convolution blocks to output the probability values of the three style labels. The local discriminators D_{ln} , D_{le} , D_{ll} also use the PatchGAN structure.

3.3.3 Drawing to Face Photo Generator F and Photo discriminator $D_{\mathcal{P}}$

The generator F in the inverse direction takes a portrait line drawing d as input and outputs a face photo $F(d)$. It uses an encoder-decoder architecture with nine remaining blocks in the middle. The photo discriminator $D_{\mathcal{P}}$ discriminates generated face photos from real ones and also adopts the PatchGAN structure.

3.3.4 Loss Functions

There are six types of losses in Quality-Metric-guided Unpaired Portrait line Drawing Generation model training.

Adversarial loss. The adversarial loss evaluates the ability of the discriminator $D_{\mathcal{D}}$ to assign correct labels to real and synthesized drawings. It is formulated as:

$$\begin{aligned} L_{adv}(G, D_{\mathcal{D}}) = & \sum_{D \in D_{\mathcal{D}}} \mathbb{E}_{d \in S(d)} [\log D(d)] \\ & + \sum_{D \in D_{\mathcal{D}}} \mathbb{E}_{p \in S(p)} [\log(1 - D(G(p, s))] \end{aligned} \quad (8)$$

where s is randomly chosen from the style features of the drawings in $S(d)$ for each p . Since $D_{\mathcal{D}}$ maximizes this loss and G minimizes it, this loss drives the generated drawings closer to real drawings.

An adversarial loss for the photo discriminator $D_{\mathcal{P}}$ and the inverse mapping F is:

$$\begin{aligned} L_{adv}(F, D_{\mathcal{P}}) = & \mathbb{E}_{p \in S(p)} [\log D_{\mathcal{P}}(p)] \\ & + \mathbb{E}_{d \in S(d)} [\log(1 - D_{\mathcal{P}}(F(d)))] \end{aligned} \quad (9)$$

Relaxed forward cycle-consistency loss. As mentioned earlier, there is much less information in the domain \mathcal{D} than in the domain \mathcal{P} . It is infeasible for $p \rightarrow G(p, s) \rightarrow F(G(p, s))$ to be pixel-wise similar to p , but the edge information in p and $F(G(p, s))$ needs to be similar, which is achievable. Edges are extracted from p and $F(G(p, s))$ using HED [46], and the similarity of edges is evaluated by the LPIPS perceptual metric [55]. Using H to denote the HED and L_{lpips} to denote the perceptual metric, the relaxed cycle consistency loss is formulated as:

$$L_{relaxed-cyc}(G, F) = \mathbb{E}_{p \in S(p)}[L_{lpips}(H(p), H(F(G(p, s))))] \quad (10)$$

Strict backward cycle-consistency loss. On the other hand, the information in the generated face photo is sufficient to reconstruct the drawing. Therefore, it is important that $d \rightarrow F(d) \rightarrow G(F(d), s(d))$ is pixelwise similar to d , where the style feature $s(d)$ is the style feature of d . The strict cycle consistency loss in the backward cycle is then formulated as:

$$L_{strict-cyc}(G, F) = \mathbb{E}_{d \in S(d)}[||d - G(F(d), s(d))||_1] \quad (11)$$

Truncation loss. The truncation loss is designed to prevent the generated drawing from hiding information in small values. It has the same format as the relaxed cycle-consistency loss, except that the generated drawing $G(p, s)$ is first truncated to 6 bits (a general digital image stores intensity in 8 bits) to ensure that the encoded information is clearly visible, and then fed into F to reconstruct the photo. Denoting the truncation operation as $T[\cdot]$, the truncation loss is formulated as:

$$L_{trunc}(G, F) = \mathbb{E}_{p \in S(p)}[L_{lpips}(H(p), H(F(T[G(p, s)])))] \quad (12)$$

During the first training period, the weight for the truncation loss is kept low, otherwise it would be too hard for the model to optimize. The weight is gradually increased as the training progresses.

Style loss. Style loss is introduced to help G generate multiple styles with different style properties. Denoting the classification branch in $D_{\mathcal{D}}$ as D_{cls} , the style loss is formulated as:

$$\begin{aligned} L_{cls}(G, D_{\mathcal{D}}) &= \mathbb{E}_{d \in S(d)}[- \sum_c p(c) \log D_{cls}(c|d)] \\ &\quad + \mathbb{E}_{p \in S(p)}[- \sum_c p'(c) \log D_{cls}(c|G(p, s))] \end{aligned} \quad (13)$$

For a real drawing d , $p(c)$ is the probability of the style label c given by the classifier C , and $D_{cls}(c|d)$ is the maximum softmax probability of D_{cls} prediction for c . The probability $p(c)$ is used to account for real drawings that may not belong to a single style but lie between two styles, e.g., the softmax probability [0.58, 0.40, 0.02]. For the generated drawing $G(p, s)$, $p'(c)$ denotes the probability on the style label c , specified by the style feature s , and $D_{cls}(c|G(p, s))$ is the softmax probability

predicted on c . This classification loss motivates D_{cls} to classify the drawing into the correct style and motivates G to generate a drawing close to the given style feature.

Quality loss based on the quality metric model. The quality loss is designed for generating high quality APDrawings. The quality metric model M gives a quality score ($\in [0.1, 1]$) of an APDrawing about how consistent it is with human perception, where better looking drawings get higher prediction scores. The quality loss $L_{quality}$ is then defined as:

$$L_{quality}(G) = \mathbb{E}_{p \in S(p)}[1 - M(G(p, s))]. \quad (14)$$

4 Recent Developments

Multi-scale methods are popular for finding the relation between the source domain and the target domain. For face photo to sketch transfer, MvDT [26] relies on a multiview domain translation method to bridge the domain discrepancy between an input test image in the source domain and a collection of images in the target domain, which flexibly integrates a Convolutional Neural Network (CNN) representation with hand-crafted features in an optimal way. Duan et al. [10] propose a multi-scale gradient self-attention residual learning framework for face photo-sketch transformation, and their method utilizes the relationship between features to selectively enhance the characteristics of specific information through self-attention distribution.

3D shapes are used in some works for sketch drawing. Neural Contours [23] is proposed for learning to generate line drawings from 3D models, and the network of Neural Contours incorporates a differentiable module operating on geometric features of the 3D model and an image-based module operating on view-based shape representations. Neural Strokes [22] takes a 3D shape and a viewpoint as input, and outputs a drawing with textured strokes, with variations in stroke thickness, deformation, and color learnt from an artist’s style.

Network architecture is another important factor for high-quality generation. Sketch-Transformer [58] contains a multi-scale feature and position encoder for a patch-level feature and position embedding, a self-attention module for capturing long-range spatial dependency, and a multi-scale spatially-adaptive de-normalization decoder for image reconstruction. CA-GAN (Composition-Aided GAN) [53] utilises paired inputs, including a face photo/sketch and the corresponding pixelwise face labels for generating a sketch/photo and stacked CA-GANs (SCA-GANs) to further rectify defects and add compelling details.

Sometimes, we would like the model to be controlled by some conditions. Wang et al. [43] propose a GAN transfer method that depends on the user input sketches. SoftGAN [5] decouples the latent space of portraits into a geometry space and a texture space, therefore it can generate high-quality portrait images with independently controllable geometry and texture attributes.

Recently, with the development of cross-modal methods, text information is embedded into visual generation models. CLIPasso [42] utilises CLIP [27], a joint image and text model, to distill semantic concepts from sketches and images alike, defines a sketch as a set of Bézier curves and uses a differentiable rasteriser to optimise the parameters of the curves directly with respect to a CLIP-based perceptual loss. CLIPasso can generalize to various categories and cope with challenging levels of abstraction while maintaining the semantic visual clues that allow for instance-level and class-level recognition. CLIPascene [41] further improves the CLIP-based method, and converts a given scene image into a sketch using different types of abstraction (precise to loose) and multiple levels of abstraction (detailed to sparse). Chan et al. [4] think that line drawings are encodings of scene information, and they propose a geometry loss to convey 3D shape and a semantic loss to match the CLIP features of a line drawing with its corresponding photograph.

5 Evaluation

Evaluation of artistic portrait drawings, such as those described in this chapter, is obviously important, but it is a challenging task. Whereas tasks such as image classification and object detection have large, annotated benchmark datasets and various natural and effective evaluation metrics (e.g. accuracy), this is largely absent from APDrawings. Probably this is a consequence of the lack of unique ground truth for image stylisation or image generation; even for a single basic style (e.g. APDrawing) there are many possible drawings that would be considered by experts to be acceptable. In contrast, for tasks such as image classification and object detection their ground truth values, at least to a first order approximation, are expected to be unique and well defined. Thus their metrics can rely on simple techniques such as counting the proportion of correct decisions. However, for APDrawings evaluation involves aesthetics, which makes it complex and difficult to measure, and is moreover subjective.

Consequently, a popular approach to performing APDrawing evaluation is to carry out a user study where the task is to assign a rating to an image or indicate a preference between several images (e.g. a two-alternative forced choice). Since carrying out such user studies is time consuming, and is also not exactly repeatable, a recent alternative has become popular: the use of the Frechet Inception distance (FID) [16] which is computed as the distance between the distribution of Inception feature vectors extracted from two sets of images. Alternatively a single image FID version [35] is available that is applied to internal patch statistics of the image. However, FID does not capture the quality of content preservation achieved by a stylisation, and moreover, is biased [7]. In addition, differences between Inception feature vectors do not always reflect human perception, although retraining the Inception network on an art dataset shows significant improvement [45].

Another approach is the ‘deception score’ which measures the proportion of stylised images classified by a VGG network as being artworks of the artist for which the stylisation was produced [34]. However, this means that its application is limited to cases in which a style is tightly specified (such that a style classifier can be trained). Also, like FID it ignores content preservation.

The NPRportrait benchmark v1.0 [32] took a different approach for evaluating content preservation. Gender, age, and ethnicity were considered to be basic features to describe faces, and, along with attractiveness, it was expected that good stylisations could preserve these characteristics unless the styles were highly abstracted. The four characteristics were estimated from the source images by a user study and taken as as ground truth, while a subsequent user study estimated the characteristics from stylised versions of the images. The distances between these distributions were taken as an indication of content loss.

NPRportrait v1.0 [32] structured the benchmark images into three levels of difficulty. As the levels increased, elements such as lighting, pose, expression, etc. were less constrained. This enabled the robustness of the stylisation algorithms to be tested by measuring their performance (e.g. according to user studies, FID, etc.) across the three levels.

6 Conclusions

This chapter focuses on the genre of portrait line drawings, and is therefore circumscribed both by medium (typically pen or pencil) and topic (typically human faces, although some artists specialise in non-humans). Nevertheless, the topic has been of interest in various forms over a long history. We describe various approaches to generating line portraits, including early work from the non-photorealistic rendering field, as well as APDrawingGAN and QMUPD, which are deep generative models. Present methods have addressed some of the challenges of line portrait generation, and achieved decent results. APDrawingGAN realises high-quality drawing with several generators for different face regions. QMUPD uses the asymmetric cycle mapping to train the generator with unpaired data, further allowing more diverse line drawing results to be learnt.

A good portrait should do more than just record a person’s physical features. An expert artist can also use a portrait to reveal the subject’s character, personality, social status, and so on [44]. Furthermore, the artist may be required to present a certain image of the subject, e.g. to idealise (or even caricature) them, send a political message, etc. In fact, traditional portraiture involves many subtleties; on one hand it aimed to provide a generic view of the sitter, but in contrast to this, also to feature the individual’s distinguishing characteristics [25]. Also, historical portraits follow many conventions regarding pose, dress, etc. and make use of emblems and symbols to communicate additional information that goes beyond mere likeness [44]. In the other direction, modernist portraiture such as Picasso or Miró involve such distortions and

abstractions that literal likeness is mostly lost, and the artist has to rely on other means to capture the person's identity.

Most of this is currently beyond the capabilities of computerised art. However, strong progress has been made in recent years, and some work has succeeded in capturing or incorporating aspects such as identity [51], semantics and 3D [4], and emotion [8]. We are happy to see more exploration in this area and hope that it will support the development of AI-Generated Content (AIGC).

Acknowledgements Ran Yi was supported by Shanghai Sailing Program (22YF1420300), CCF-Tencent Open Research Fund (RAGR20220121), Young Elite Scientists Sponsorship Program by CAST (2022QNRC001), National Natural Science Foundation of China (62272447), Beijing Natural Science Foundation (L222117).

References

1. Anoosheh, A., Agustsson, E., Timofte, R., & Gool, L. V. (2018). ComboGAN: unrestrained scalability for image domain translation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, pp. 783–790.
2. Baluja, S. (2022). A natural representation of colors with textures. *The Visual Computer*, 38(9), 3267–3278.
3. Berger, I., Shamir, A., Mahler, M., Carter, E., & Hodgins, J. (2013). Style and abstraction in portrait sketching. *ACM Transactions on Graphics (TOG)*, 32(4), 55.
4. Chan, C., Durand, F., & Isola, P. (2022). Learning to generate line drawings that convey geometry and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7915–7925).
5. Chen, A., Liu, R., Xie, L., Chen, Z., Su, H., & Yu, J. (2022). SofGAN: A portrait image generator with dynamic styling. *ACM Transactions on Graphics (TOG)*, 41(1), 1–26.
6. Chiu, C. C., Lo, Y. H., Lee, R. R., & Chu, H. K. (2015). Tone-and feature-aware circular scribble art. In *Computer Graphics Forum* (Vol. 34, pp. 225–234).
7. Chong, M. J., & Forsyth, D. (2020) Effectively unbiased FID and inception score and where to find them. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6070–6079).
8. Colton, S., Valstar, M. F., & Pantic, M. (2008). Emotionally aware automated portrait painting. In *Proceedings of the 3rd International Conference on Digital Interactive Media in Entertainment and Arts* (pp. 304–311).
9. Cootes, T. F., Edwards, G. J., & Taylor, C. J. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6), 681–685.
10. Duan, S., Chen, Z., Wu, Q. J., Cai, L., & Lu, D. (2020). Multi-scale gradients self-attention residual learning for face photo-sketch transformation. *IEEE Transactions on Information Forensics and Security*, 16, 1218–1230.
11. Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)* (pp. 2414–2423).
12. Gooch, B., Reinhard, E., & Gooch, A. (2004). Human facial illustrations: Creation and psychophysical evaluation. *ACM Transactions on Graphics*, 23(1), 27–44.
13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS '14)* (pp. 2672–2680).

14. Gu, S., Bao, J., Yang, H., Chen, D., Wen, F., & Yuan, L. (2019). Mask-guided portrait editing with conditional GANs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3436–3445).
15. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778).
16. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in neural information processing systems* (Vol. 3).
17. Huang, R., Zhang, S., Li, T., & He, R. (2017). Beyond face rotation: Global and local perception GAN for photorealistic and identity preserving frontal view synthesis. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV '17)* (pp. 2439–2448).
18. Isola, P., Zhu, J., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '17)* (pp. 1125–1134).
19. Kang, H., Lee, S., & Chui, C. K. (2007). Coherent line drawing. In *ACM Symposium Non-photorealistic Animation and Rendering* (pp. 43–50).
20. Li, C., & Wand, M. (2016) Combining markov random fields and convolutional neural networks for image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)* (pp. 2479–2486).
21. Liao, J., Yao, Y., Yuan, L., Hua, G., & Kang, S. B. (2017). Visual attribute transfer through deep image analogy. *ACM Transactions on Graphics (TOG)*, 36(4), 120:1–120:15.
22. Liu, D., Fisher, M., Hertzmann, A., & Kalogerakis, E. (2021). Neural strokes: Stylized line drawing of 3d shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 14204–14213).
23. Liu, D., Nabail, M., Hertzmann, A., & Kalogerakis, E. (2020). Neural contours: Learning to draw lines from 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5428–5436).
24. Meng, M., Zhao, M., & Zhu, S. C. (2010). Artistic paper-cut of human portraits. In *Proceedings of the 18th International Conference on Multimedia* (pp. 931–934). ACM.
25. Panofsky, E. (2019). *Early Netherlandish painting: Its origins and character*. Routledge.
26. Peng, C., Wang, N., Li, J., & Gao, X. (2020). Universal face photo-sketch style transfer via multiview domain translation. *IEEE Transactions on Image Processing*, 29, 8519–8534.
27. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning* (pp. 8748–8763). PMLR.
28. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI '15)* pp. 234–241.
29. Rosin, P. L., & Lai, Y. K. (2013). Artistic minimal rendering with lines and blocks. *Graphical Models*, 75(4), 208–229.
30. Rosin, P. L., & Lai, Y. K. (2015). Non-photorealistic rendering of portraits. In *Proceedings of the Workshop on Computational Aesthetics* (pp. 159–170). Eurographics Association (2015).
31. Rosin, P. L., & Lai, Y. K. (2020). *Image-based portrait engraving*. arXiv preprint [arXiv:2008.05336](https://arxiv.org/abs/2008.05336).
32. Rosin, P. L., Lai, Y. K., Mould, D., Yi, R., Berger, I., Doyle, L., Lee, S., Li, C., Liu, Y. J., Semmo, A., et al. (2022). NPRportrait 1.0: A three-level benchmark for non-photorealistic rendering of portraits. *Computational Visual Media*, 8(3), 445–465.
33. Rother, C., Kolmogorov, V., & Blake, A. (2004). “GrabCut”: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3), 309–314.
34. Sanakoyeu, A., Kotovenko, D., Lang, S., & Ommer, B. (2018). A style-aware content loss for real-time HD style transfer. In *European Conference on Computer Vision* (pp. 698–714).
35. Shaham, T. R., Dekel, T., & Michaeli, T. (2019). SinGAN: Learning a generative model from a single natural image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 4570–4580).

36. Shen, X., Hertzmann, A., Jia, J., Paris, S., Price, B., Shechtman, E., & Sachs, I. (2016). Automatic portrait segmentation for image stylization. *Computer Graphics Forum*, 35(2), 93–102.
37. Shih, Y., Paris, S., Barnes, C., Freeman, W. T., & Durand, F. (2014). Style transfer for headshot portraits. *ACM Transactions on Graphics (TOG)*, 33(4), 148:1–148:14.
38. Son, M., Lee, Y., Kang, H., & Lee, S. (2011). Structure grid for directional stippling. *Graphical Models*, 73(3), 74–87.
39. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1), 1929–1958.
40. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2818–2826).
41. Vinker, Y., Alaluf, Y., Cohen-Or, D., & Shamir, A. (2022). CLIPascene: Scene sketching with different types and levels of abstraction. arXiv preprint [arXiv:2211.17256](https://arxiv.org/abs/2211.17256).
42. Vinker, Y., Pajouheshgar, E., Bo, J. Y., Bachmann, R. C., Bermano, A. H., Cohen-Or, D., Zamir, A., & Shamir, A. (2022). CLIPasso: Semantically-aware object sketching. *ACM Transactions on Graphics (TOG)*, 41(4), 1–11.
43. Wang, S. Y., Bau, D., & Zhu, J. Y. (2021). Sketch your own GAN. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 14050–14060).
44. West, S. (2004). *Portraiture*. Oxford: Oxford University Press.
45. Wright, M., & Ommer, B. (2022). ArtFID: Quantitative evaluation of neural style transfer. In *Proceedings of 44th DAGM German Conference* (pp. 560–576).
46. Xie, S., & Tu, Z. (2015). Holistically-nested edge detection. In *IEEE International Conference on Computer Vision (ICCV)* (pp. 1395–1403).
47. Yi, R., Liu, Y., Lai, Y., & Rosin, P. L. (2020). Unpaired portrait drawing generation via asymmetric cycle mapping. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 8214–8222).
48. Yi, R., Liu, Y., Lai, Y., & Rosin, P. L. (2023). Quality metric guided portrait line drawing generation from unpaired training data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1), 905–918.
49. Yi, R., Liu, Y. J., Lai, Y. K., & Rosin, P. L. (2019). APDrawingGAN: Generating artistic portrait drawings from face photos with hierarchical GANs. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 10743–10752).
50. Yi, R., Xia, M., Liu, Y., Lai, Y., & Rosin, P. L. (2021). Line drawings for face portraits from photos using global and local structure based GANs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10), 3462–3475.
51. Yi, R., Ye, Z., Fan, R., Shu, Y., Liu, Y. J., Lai, Y. K., & Rosin, P. L. (2022). Animating portrait line drawings from a single face photo and a speech signal. In *ACM SIGGRAPH 2022 Conference Proceedings* (pp. 1–8).
52. Yi, Z., Zhang, H. R., Tan, P., & Gong, M. (2017). DualGAN: Unsupervised dual learning for image-to-image translation. In *IEEE International Conference on Computer Vision (ICCV)* (pp. 2868–2876).
53. Yu, J., Xu, X., Gao, F., Shi, S., Wang, M., Tao, D., & Huang, Q. (2020). Toward realistic face photo-sketch synthesis via composition-aided GANs. *IEEE Transactions on Cybernetics*, 51(9), 4350–4362.
54. Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10), 1499–1503.
55. Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 586–595).
56. Zhu, J., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)* (pp. 2242–2251).

57. Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV '17)* (pp. 2223–2232).
58. Zhu, M., Liang, C., Wang, N., Wang, X., Li, Z., & Gao, X. (2021). A sketch-transformer network for face photo-sketch synthesis. In *IJCAI* (pp. 1352–1358).

AI Deep Learning Generative Models for Drug Discovery



Qifeng Bai, Jian Ma, and Tingyang Xu

Abstract Artificial intelligence (AI) deep learning generative models play an increasingly important role in drug design. Developments of different drug generative models can save capital and time to promote new drug discovery. AI deep learning generative models can be divided into different generative models based on the different levels of dimensional features of receptors and ligands such as SMILES generative models, molecular graph generative models, and 3D molecule generative models. Besides, based on the different algorithms, AI deep learning generative models for drug discovery can be roughly classified as variational autoencoder generative model, generative adversarial network generative model, and flow based generative model, and diffusion generative model. In this chapter, the classification, general mathematical methods, and research reports of AI deep learning generative models are summarized based on the different levels of dimensional features and algorithms. This chapter proposes an interesting topic and a deep understanding of AI deep learning generative models for the scientific community.

Keywords Artificial intelligence · AI · Deep learning · Generative model · Drug discovery

1 Introduction

Generative models based on artificial intelligence (AI) deep learning have become a popular and effective way to design new novel drugs. Generative models for drug design have gone through different stages. The quantitative structure–activity relationship (QSAR) modeling was proposed in the 1930s to study the mathematical relation between drug structures and properties that fitted into the chemical or biological

Q. Bai (✉) · J. Ma

School of Basic Medical Sciences, Lanzhou University, Lanzhou 730000, Gansu, People's Republic of China
e-mail: baiqf@lzu.edu.cn

T. Xu

Tencent AI Lab, Shenzhen, People's Republic of China

activities that can be further used for drug virtual screening [1]. With the development of computational methods, the de novo drug design methods are devised to generate new novel molecules with the desired biochemical attributes [2]. Some academic soft packages such as MolAICal [3], LigBuilder [4], and OpenGrowth [5] are written for molecular generation research based on 3D receptor pockets. In particular, MolAICal can employ the deep learning generative model and classical algorithm to carry out the fragment generation on the initial setting structure in the 3D pocket of receptors, which indicates that deep learning generative models have a promising application in drug discovery [6, 7]. Deep learning methods supply an automatic way to deal with the big challenge problems that are involved in extracting the desired drug features and searching the large sample space along with the discontinuous natures of the drug optimization domain [8–10]. Currently, drug generative models are mainly trained based on simplified molecular-input line-entry system (SMILES) [11] and molecular graphs [12]. SMILES generation models are easy to be trained like the language model based on 1D sequence characters. However, SMILES is hard to express the essential chemical and physical properties such as 3D geometrical coordinates so that the drug generative model cannot produce the molecules with 3D conformations [13]. Molecular graph methods can compensate for SMILES shortcomings to train the drug 3D conformation models [14]. Due to the equivariant symmetries in the operations of rotations, translations, and reflections on the drugs, Equivariant Graph Neural Networks (EGNNs) [15] are proposed for dealing with the high dimensional information data.

With the development of deep learning, there has been an increasing use of generative models for molecular generation. The process of molecule generation can be mainly categorized into three methods: atom-by-atom, fragment-by-fragment, or direct generation of complete molecules. From a design dimensional perspective, molecular generation models can be classified into 1D SMILES generation models, 2D molecular graph generation models, and 3D molecular geometry generation models. Besides, there are four common kinds of drug generative models that include variational autoencoder (VAE) generative model [16], generative adversarial network (GAN) generative model [17], flow based generative model [18], and diffusion generative model [19]. This chapter provides a relatively in-depth introduction to the classification, general mathematical methods, and research reports of AI deep learning generative models.

2 Different Hierarchical Generative Models for Drug Discovery

2.1 SMILES Generative Models

The SMILES notation is a natural language-like format used for representing molecules and generating novel molecules using SMILES models has become an interesting area for numerous researchers in the field of drug discovery. A recurrent neural network (RNN) was proposed by [20] for de novo drug design, and the molecule generation process can be described by following Eq. 1:

$$P_{\theta}(S) = P_{\theta}(s_1) \cdot \prod_{t=2}^T P_{\theta}(s_t | s_{t-1}, \dots, s_1) \quad (1)$$

where the S is the generated sequence and s_i is the symbols at steps t , and the P_{θ} is the trained model. To improve the quality of generated sequences, the model incorporates a long short-term memory (LSTM) architecture. Inspired by the success of the generative pre-training models (GPT), Bagal et al. [21] trained a model named MolGPT for the generation of druglike molecules, and the model is the mini version of GPT1. MolGPT's architecture is composed of stacked decoder blocks, and each contains a fully connected network and a masked self-attention layer. To preserve the input sequence order, position value embeddings for each SMILES token are incorporated into the model inputs. Once training is finished, a start token is fed to the model to begin the molecule generation process, wherein subsequent tokens are generated sequentially, and eventually, a complete molecule is sampled.

2.2 Molecular Graph Generative Models

The use of graphs to represent the topological structure of molecules by nodes and edges, as well as to extract node-to-node relationships, has become increasingly prevalent in research projects aimed at generating molecular graphs. Jin et al. [22] aimed to leverage the power of variational autoencoder for molecular graph generation, developing a novel model known as the junction tree variational autoencoder (JT VAE) in their work. They thought that the node-by-node generation method for molecular graphs was not ideal, as it could lead the model to generate chemically invalid intermediate molecules, potentially affecting the final validation of the complete generated molecules. Therefore, they developed a new approach exploiting valid subgraphs as components to generate the complete molecular graph in two steps. The generation process involves using the tree decoder to first generate the junction tree of a molecule from a latent vector sampled from a normal distribution, which is then utilized by the graph decoder to construct the correct molecular graph through

node assembly in the tree structure. Jin et al. [23] also proposed a model that employs structural motifs to generate molecular graphs hierarchically. Their model employs a joint distribution over structural motifs to model the probability of a molecular graph \mathcal{G} , resulting in an auto-regressive factorization of the probability $P(\mathcal{G})$ (see Eq. 2).

$$P(\mathcal{G}) = \int P(z) \prod_k P(S_k, A_k | S_{<k}, A_{<k}, z) dz \quad (2)$$

where S_k is the new structural motif, and A_k is the attachment indicating the intersecting atoms between a motif and its neighbor motifs. The basic architecture of the motif model is the variational autoencoder containing the encoder and decoder. The encoder is composed of atom, attachment, and motif layers used to encode the features of molecular graphs at different levels. The decoder generates molecular graphs by performing three steps in a specific order: motif prediction, attachment prediction, and graph prediction. This process involves decoding a latent variable to generate a structural motif, which is then used to predict the attachments between motifs, and finally, to combine these attachments to form a full molecular graph. Shi et al. [24] introduced a model called GraphAF which was a flow-based autoregressive model for graph generation. A normalizing flow can transform a base distribution \mathcal{E} (e.g., Gaussian distribution) to real-world data distribution \mathcal{Z} (e.g., language and compounds), which can be described by the change-of-variables Eq. 3:

$$p_{\mathcal{Z}}(z) = p_{\mathcal{E}}(f_{\theta}^{-1}(z)) \left| \det \frac{\partial f_{\theta}^{-1}(z)}{\partial z} \right| \quad (3)$$

where parameter z represents the node type X_i or edge type A_{ij} . Because the node and edge type are both discrete, the dequantization technique [25] is leveraged to add the real-world noise to convert discrete data into continuous data, and the approach can be represented by this Eq. 4:

$$z_i^X = X_i + \mu, \quad \mu \sim U[0, 1]^d; \quad z_{ij}^A = A_{ij} + \mu, \quad \mu \sim U[0, 1]^{b+1} \quad (4)$$

The generation procedure of GraphAF starts from an empty graph and iteratively generates atoms and bonds by sample random noise. Despite the aforementioned limitation of this atom-by-atom approach, the authors employed existing chemical rules to verify whether the current bonds exceeded the allowed valency at each generation step, guaranteeing the validity of the eventual molecular graphs. Vignac et al. [26] proposed the Discrete Graph Denoising Diffusion model (DiGress) that was used to generate molecular graphs with categorical node and edge attributes. DiGress directly generates an entire graph recursively using a diffusion network, which differs from the sequential graph generation method. Besides, a conditional generation method has been proposed based on the well-trained DiGress model, which was used to generate graphs with desired properties.

2.3 3D Molecule Generative Models

3D molecule generation models which need to consider the higher dimensional features such as 3D coordinates, translational, rotational, and torsional spaces, are more in line with the requirements of reasonable structure-based drug design. There have been some research reports about 3D molecule generation models. For instance, Ragoza et al. [27] present a generative model generating 3D molecules conditional on the target pocket, and the model is built based on a conditional variational autoencoder (CVAE). They assume that the complex of protein and ligand can be modeled as a latent variable z , which follows a prior distribution representing binding interaction, such as Gaussian distribution. The generative model architecture consists of two encoders and one decoder: one encoder is an input encoder encoding the protein–ligand complex, while the other encoder is a conditional encoder encoding the pocket. Both encoders encode the format of molecules in atomic density grids. The decoder then utilized this information to output an atomic density grid, which was ultimately processed by the atom fitting algorithm to convert it into a discrete 3D molecular structure. Luo et al. [28] proposed an autoregressive normalizing flow model for 3D molecular generation (G-SphereNet). Their molecular generative method works atom-by-atom, where the type and position of the next atom are determined based on the previously generated ones. Designing a flow model that directly generates atom coordinates while simultaneously satisfying the properties of invariance and equivariance in the generation process is challenging; therefore, their model does not directly generate the coordinates of the next atom. Instead, it employs the coordinates of the already generated atoms to generate the distance, angle, and torsion, which then allow for the calculation of the coordinates of the next atom. The 3D point cloud $G = (A, R)$ was used to represent the geometry structure of a molecule, so their generation process can be described by the Eq. 5:

$$a_i = g^a(z_i^a; A_i, R_i), \quad r_i = g^r(z_i^r; A_i, R_i), \quad i \geq 1 \quad (5)$$

where g^a and g^r are the atom type generation process and atom coordinates generation process, respectively. To ensure that the generated molecules bind to the targeted proteins accurately, Liu et al. [29] extended the G-SphereNet approach and introduced the GraphBP model, which could generate 3D molecules that bound to specific proteins. Although the autoregressive flow model is capable of generating molecules successfully, it has certain drawbacks that include: (1) an accumulation process in which the final outcome is dependent on potentially poor intermediate states, and (2) a limited capacity to generate larger molecules. Hoogeboom et al. [30] first employed the diffusion model for the task of molecular generation, which they named E(3) Equivariant Diffusion Models (EDM). Their model operates directly on atom coordinates and types. The training process of EDM can be divided into two processes: the forward (diffusion) process and the reverse (denoising) process. Given a data point x representing the 3D structure of molecules, the forward process involves adding

Gaussian noise step-by-step in a Markov manner until the original data (molecular structure) is destroyed, as described by the following Eq. 6:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \alpha_t x_{t-1}, \beta_t I) \quad (6)$$

where x_t represents the intermediate of noisy molecules and setting $\alpha_t = 1 - \beta_t$, and the process results in the final distribution x_T getting closer to the standard normal distribution. The reverse process aims to train a neural network to steadily remove noise from random noise and produce valid molecules. Hence, the complete denoising process can be expressed in the form of Eq. 7:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \eta_t I) \quad (7)$$

where μ_θ represents the parameterized neural network. The equivariant network EGNN [15] was employed to model the reverse process of the diffusion model to make sure the equivariance of molecules. Huang et al. [31] proposed a molecular diffusion model named MDM that divided the molecular edges into local edges and global edges and used the equivariant local encoders and the equivariant global encoders to extract features from these two types of edges. Experiments conducted on two molecule datasets, QM9 and GEOM, showed that MDM outperforms EDM. Based on MDM, they introduced the PMDM [32] model which took into account the information of the protein pocket, enabling it to generate high affinity molecules for a specific pocket. Xu et al. [33] believed that the atomic space, composed of diverse features such as atom types, charges, and coordinates, renders unified Gaussian diffusion frameworks suboptimal. To address this issue, similar to latent diffusion models (LDMs) [34] that operated a diffusion process on the latent space of high-resolution images, they presented a model called Geometric Latent Diffusion Models (GeoLDM), which run the diffusion model on a smoother latent space. Their model architecture comprises a geometric autoencoder (AE) and a geometric latent diffusion model. The training process of the model consists of two stages, where they initially train the AE, followed by the training of the latent diffusion model on the latent embedding encoder using the pre-trained encoder. Since the encoder is not utilized for molecule sampling due to the model's specificity in architecture, the inference process exclusively involves a denoising loop in the latent space, and the final decoding of the atom space is executed by the decoder. Zhang et al. [35] proposed a fragment-by-fragment framework called FLAG to directly generate 3D molecules, which differed from 2D fragment-based molecular graph generation. During the generation procedure, the pocket information is also considered. If molecule can be represented as \mathcal{G} and \mathcal{P} , the procedure can be summarized as Eqs. 8 and 9:

$$\mathcal{G}_t = \varphi(\mathcal{G}_t, \mathcal{P}), t > 1 \quad (8)$$

$$\mathcal{G}_1 = \varphi(\mathcal{P}), t = 1 \quad (9)$$

In Eq. 8, \mathcal{G}_t refers to the intermediate molecules generated at step t. Compared to methods that immediately generate entire molecules, their model employed a sequential generation process consisting of focal motif selection, next motif prediction, motif attachment enumeration and prediction, rotation angle prediction, and structure refinement. This iterative process is repeated until no additional motifs or atoms can be selected as the focal point, at which point molecule generation is considered complete.

3 Deep Generative Models for Drug Discovery

3.1 Variational Autoencoder Generative Model

Variational autoencoder (VAE) is a neural network architecture that encodes data into a low-dimensional representation and outputs reconstructed samples that are similar to the original input dataset based on probabilistic graphical models and variational Bayesian methods [16]. As shown in Fig. 1, the original input data x is compressed into the latent encoding vector z via the encoder network, the output is the reconstructive data that are recovered from the latent encoding vector Z . The data generation function for the encoding vector is defined as Eq. 10:

$$p_{\theta}(x^{(i)}) = \int p_{\theta}(x^{(i)}|z) p_{\theta}(z) dz \quad (10)$$

where $z^{(i)}$ is sampled from a prior distribution $p_{\theta^*}(z)$ and $x^{(i)}$ is produced by a conditional distribution $p_{\theta^*}(x|z = z^{(i)})$. θ^* is an optimal parameter which is defined as Eq. 11:

$$\theta^* = \arg \max_{\theta} \sum_{i=1}^n \log p_{\theta}(x^{(i)}) \quad (11)$$

Due to the difficult computation of $p_{\theta}(x^{(i)})$, the approximation function $q_{\phi}(z|x)$ is proposed to quantify the distance with the real one $p_{\theta}(z|x)$ by Kullback–Leibler divergence. The loss function of VAE is defined as Eq. 12:

$$\begin{aligned} L_{\text{VAE}}(\theta, \phi) &= -\log p_{\theta}(x) + D_{\text{KL}}(q_{\phi}(z|x) \| p_{\theta}(z|x)) \\ &= -\mathbb{E}_{z \sim q_{\phi}(z|x)} \log p_{\theta}(x|z) + D_{\text{KL}}(q_{\phi}(z|x) \| p_{\theta}(z)) \end{aligned} \quad (12)$$

To make the generated samples from $z \sim q_{\phi}(z|x)$ trainable, the reparameterization trick is used as Eq. 13:

$$z = \mu + \sigma \odot \varepsilon, \quad \text{where } \varepsilon \sim \mathcal{N}(0, I) \quad (13)$$

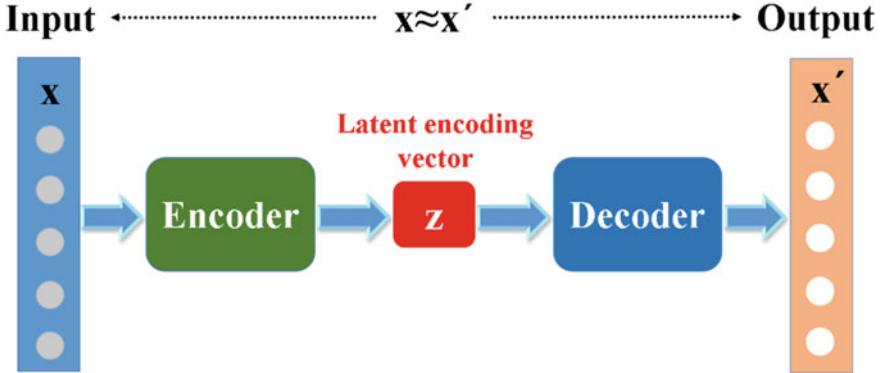


Fig. 1 Variational autoencoder architecture

μ and σ represent mean and variance that can be learned during the model training process. The VAE methods have been reported for drug design. For instance, 3DLinker [36] can predict the jointly generate linker graphs and anchor atoms for 3D structural generation based on the E(3) equivariant graph variational autoencoder that can avoid atom clashes.

3.2 Generative Adversarial Network Generative Model

Generative adversarial network (GAN) [17] has been widely used in many generative tasks such as drugs, images, and music. It consists of two parts: a generator G that tries to produce the synthetic samples as real as possible to fake the discriminator, and a discriminator D that tries to distinguish the fake samples from the real dataset. The generator G and discriminator D are competing together while making each other better (see Fig. 2). D and G play a minimization and maximization game for model training by the following loss function (see Eq. 14):

$$\min_G \max_D L(D, G) = \mathbb{E}_{x \sim p_r(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (14)$$

The discriminator D tries to identify real data as much as possible by maximizing $\mathbb{E}_{x \sim p_r(x)} [\log D(x)]$. At the same time, the discriminator D is also expected to let $D(G(z))$ close to zero. In contrast, the generator G tries to increase the probability of discriminator D for the fake samples by minimizing $\mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$.

However, GAN faces the problems of convergence [37], vanishing gradient [38], instability [38], mode collapse [39], and so on. There are some solutions for these problems of GAN such as minibatch discrimination, historical averaging, feature matching, and Wasserstein distance [40]. Wasserstein distance which is also called

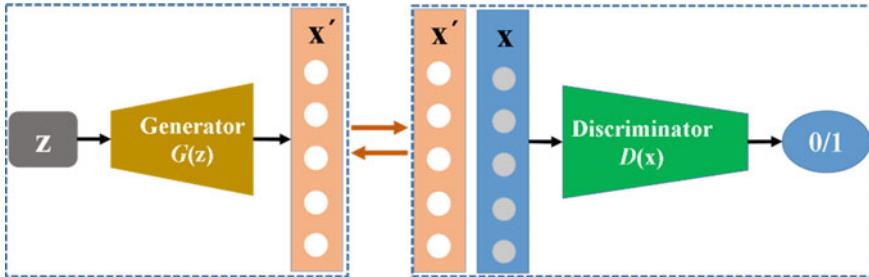


Fig. 2 GAN adversarial training architecture

Earth Mover's distance can not only be used to measure the distance between two probability distributions, but also can supply a smooth representation of the distance between two probability distributions [40]. Wasserstein distance is used to improve the GAN loss as shown in Eq. 15:

$$\begin{aligned} L(p_r, p_g) &= W(p_r, p_g) \\ &= \max_{w \in W} \mathbb{E}_{x \sim p_r}[f_w(x)] - \mathbb{E}_{z \sim p_r(z)}[f_w(g_\theta(z))] \end{aligned} \quad (15)$$

where f is K -Lipschitz function. In Wasserstein GAN, the discriminator does not directly tell the real dataset apart from the generated fake samples. Instead, Wasserstein distance is trained between the data distribution of the generated fake samples and the real dataset based on the K -Lipschitz continuous function (see Eq. 15). If the Wasserstein distance becomes smaller, the output of the generator will be close to the real dataset distribution. The GAN models are widely used for drug discovery. For example, MolAICal [3, 41] is designed for 3D drug generation in the 3D protein pocket based on the genetic algorithm and GAN models that are trained on the FDA-approved drug fragment dataset.

3.3 Flow Based Generative Model

Unlike the generative models of the generative adversarial network (GAN) [17] and variational autoencoder (VAE) [42], the flow-based deep generative models such as the normalizing flow model can explicitly learn the probability density function of all possible values of real data via building a sequence of invertible transformations (see Fig. 3a). Several normalizing flow models such as RealNVP [43], Glow [44], and MAF [45] exhibit a good and powerful distribution approximation to transform a simple distribution into the complicated one by operating a sequence of invertible transformations. Similar equation formats are used in flow-based models as Eqs. 16 and 17:

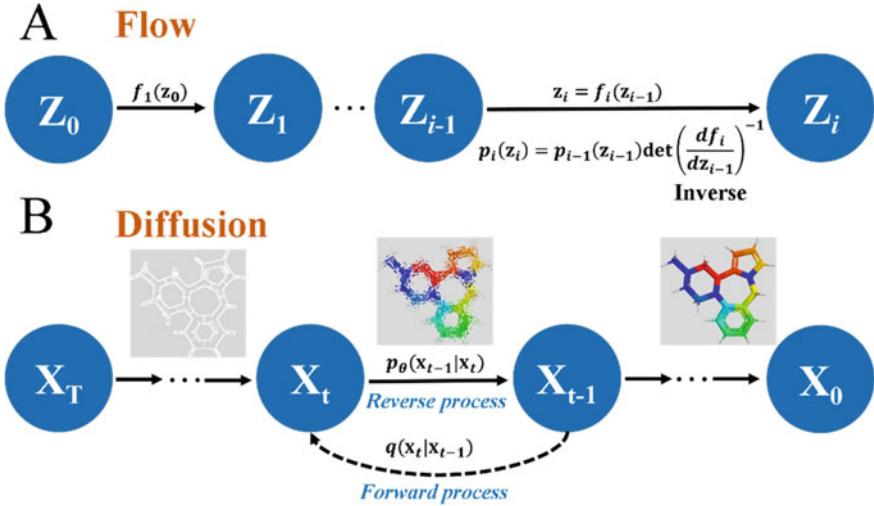


Fig. 3 (a) Flow based model and (b) diffusion model

$$z \sim \pi(z), \quad x = f(z), \quad z = f^{-1}(x) \quad (16)$$

$$p(x) = \pi(z) \mid \det \frac{dz}{dx} \mid = \pi(f^{-1}(x)) \mid \det \frac{df^{-1}}{dx} \mid \quad (17)$$

It is very easy to deduce Eq. 18 for normalizing flow as shown in Fig. 3a based on Eqs. 16, 17 and the inverse function theorem:

$$p_i(z_i) = p_{i-1}(z_{i-1}) \mid \det \left(\frac{df_i}{dz_{i-1}} \right)^{-1} \mid \quad (18)$$

The normalizing flow traces back to the beginning distribution z_i from z_N via expanding the equation of output x between two consecutive variables as Eqs. 19 and 20:

$$x = z_N = f_N \circ f_{N-1} \circ \cdots \circ f_1(z_0) \quad (19)$$

$$\begin{aligned} \log p(x) &= \log \pi_N(z_N) = \log \pi_{N-1}(z_{N-1}) - \log \mid \det \frac{df_N}{dz_{N-1}} \mid \\ &= \log \pi_0(z_0) - \sum_{i=1}^N \log \mid \det \frac{df_i}{dz_{i-1}} \mid \end{aligned} \quad (20)$$

where df_i/dz_{i-1} represents the Jacobian matrix. The loss function of the flow-based generative model is optimized with the negative log-likelihood (see Eq. 21):

$$\mathcal{L}(\mathcal{D}) = -\frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \log p(x) \quad (21)$$

where \mathcal{D} is the prepared dataset for model training. The flow based model has been reported for molecular generation in some research reports. For example, FastFlows [46] which is trained for molecular graph generation based on flow based model, can fastly generate thousands of small valid molecules with the desired attributes of drug-likeness, synthetic accessibility, and synthetic complexity. Besides, GraphNVP [47] learns the latent space to generate molecules with the desired physicochemical properties by inverting the whole graph components that contain the adjacency tensors and dequantized node attributes based on two novel invertible flows.

3.4 Diffusion Generative Model

Diffusion models which are derived from the related methods of non-equilibrium thermodynamics consist of forward diffusion processes that are responsible for gradually adding random noise to the data via referring to the methods of stochastic gradient descent and stochastic gradient Langevin dynamics [48], and the reverse diffusion processes that learn the desired data samples from the generated noise in the defined Markov chain (see Fig. 3B). Currently, some excellent diffusion-based methods are reported with similar concepts such as denoising diffusion probabilistic models (DDPM) [49], diffusion probabilistic models [50], and noise-conditioned score network (NCSN) [51]. The forward process $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$ and reverse process $p_\theta(\mathbf{x}_{0:T})$ are defined as Eqs. 22 and 23:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \prod_{t=1}^T \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right) \quad (22)$$

$$\begin{aligned} p_\theta(\mathbf{x}_{0:T}) &= p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \\ &= p(\mathbf{x}_T) \prod_{t=1}^T \mathcal{N}\left(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sum_\theta (\mathbf{x}_t, t)\right) \end{aligned} \quad (23)$$

where $\{\beta_t \in (0, 1)\}_{t=1}^T$ is a variance schedule that adjusts step sizes for the generated sequence of Gaussian noise samples $\mathbf{x}_1, \dots, \mathbf{x}_T$ in T steps. Based on Eqs. 22 and 23 and Kullback–Leibler (KL) divergence, the diffusion model training is optimized by negative log-likelihood as shown in Eq. 24:

$$\begin{aligned}
-\mathbb{E}_{q(x_0)} \log p_\theta(x_0) &\leq \mathbb{E}_{q(x_{0:T})} \left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} \right] \\
&= \mathbb{E}_q \left[-\log p(x_T) - \sum_{t \geq 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} \right] \\
&= \mathbb{E}_q \left[\underbrace{D_{\text{KL}}(q(x_T|x_0) \| p_\theta(x_T))}_{L_T} + \sum_{t>1} \underbrace{D_{\text{KL}}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t))}_{L_{t-1}} \right. \\
&\quad \left. - \underbrace{\log p_\theta(x_0|x_1)}_{L_0} \right]
\end{aligned} \tag{24}$$

The final optimization function is due to minimize the comparison loss between Gaussians based on KL divergence theory (see Eq. 24). Since diffusion models can be trained with fixed steps based on high dimensional latent variables such as 3D coordinates in the original molecular data, it is considered to be suitable to learn 3D molecular docking or generation models of ligands in the 3D protein pocket. For example, DiffDock [52] employs an efficient diffusion process to generate the new translational, rotational, and torsional spaces for model training, and it shows prominently higher precision for the 3D molecular docking of drugs.

4 Discussion and Conclusions

In this chapter, an over review of AI deep learning generative models for drug discovery is represented based on the different levels of dimensional features and algorithms. The SMILES generation, molecular graph generation, and 3D molecule generation models are introduced with some research reports. Of course, the n-dimensional generative models are also trainable according to E(n)-Equivariant Graph Neural Networks (EGNNs). This chapter also includes the introduction of the common generative models such as the variational autoencoder generative model, generative adversarial network generative model, and flow based generative model. Besides, the autoregressive generative model [53] and energy-based generative model [54, 55] can also be used for drug generation. Actually, these two generation methods can be used together with the above four common generative models. For example, if each dimension of normalizing flow transformation is conditioned on the previous dimensions, this kind of flow is an autoregressive flow such as PixelRNN [56]. Energy-based generative methods can employ Markov Chain Monte Carlo (MCMC) or Hamiltonian Monte Carlo (HMC) algorithm to obtain reasonable samples which is similar to the diffusion method. Inspired by the progress of the diffusion model, the energy-based generative models are proposed based on increasingly noisy versions of a dataset via a diffusion recovery likelihood method [57]. We hope the knowledge

of this chapter can be helpful for the related researchers who are interested in deep learning generative models for drug discovery.

References

- Cherkasov, A., Muratov, E. N., Fourches, D., Varnek, A., Baskin, I. I., Cronin, M., et al. (2014). QSAR modeling: Where have you been? Where are you going to? *Journal of Medicinal Chemistry*, 57(12), 4977–5010. <https://doi.org/10.1021/jm4004285>
- Schneider, G., & Fechner, U. (2005). Computer-based de novo design of drug-like molecules. *Nature Reviews. Drug Discovery*, 4(8), 649–663. <https://doi.org/10.1038/nrd1799>
- Bai, Q., Tan, S., Xu, T., Liu, H., Huang, J., & Yao, X. (2021). MolAICal: A soft tool for 3D drug design of protein targets by artificial intelligence and classical algorithm. *Briefings in bioinformatics*, 22(3), bbaa161.
- Wang, R., Gao, Y., & Lai, L. (2000). LigBuilder: A multi-purpose program for structure-based drug design. *Molecular Modeling Annual*, 6(7), 498–516. https://doi.org/10.1007/s00894000_60498
- Cheron, N., Jasty, N., & Shakhnovich, E. I. (2016). OpenGrowth: An automated and rational algorithm for finding new protein ligands. *Journal of Medicinal Chemistry*, 59(9), 4171–4188. <https://doi.org/10.1021/acs.jmedchem.5b00886>
- Bai, Q., Ma, J., Liu, S., Xu, T., Banegas-Luna, A. J., Pérez-Sánchez, H., et al. (2021). WADDAICA: A webserver for aiding protein drug design by artificial intelligence and classical algorithm. *Computational and Structural Biotechnology Journal*, 19, 3573–3579. <https://doi.org/10.1016/j.csbj.2021.06.017>
- Bai, Q., Liu, S., Tian, Y., Xu, T., Banegas-Luna, A. J., Pérez-Sánchez, H., et al. (2022). Application advances of deep learning methods for de novo drug design and molecular dynamics simulation. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 12(3), e1581. <https://doi.org/10.1002/wcms.1581>
- Yang, S.-Q., Ye, Q., Ding, J.-J., Yin, M.-Z., Lu, A.-P., Chen, X., et al. (2021). Current advances in ligand-based target prediction. *WIREs Computational Molecular Science*, 11(3), e1504. <https://doi.org/10.1002/wcms.1504>
- Polishchuk, P. G., Madzhidov, T. I., & Varnek, A. (2013). Estimation of the size of drug-like chemical space based on GDB-17 data. *Journal of Computer-Aided Molecular Design*, 27(8), 675–679. <https://doi.org/10.1007/s10822-013-9672-4>
- Stumpfe, D., & Bajorath, J. (2012). Exploring activity cliffs in medicinal chemistry. *Journal of Medicinal Chemistry*, 55(7), 2932–2942. <https://doi.org/10.1021/jm201706b>
- Wiswesser, W. J. (1985). Historic development of chemical notations. *Journal of Chemical Information and Computer Sciences*, 25(3), 258–263. <https://doi.org/10.1021/ci00047a023>
- Wang, Y., Li, Z., & Farimani, A. B. (2022). Graph neural networks for molecules. arXiv preprint arXiv:220905582.
- Xu, Y., Lin, K., Wang, S., Wang, L., Cai, C., Song, C., et al. (2019). Deep learning for molecular generation. *Future Medicinal Chemistry*, 11(6), 567–597. <https://doi.org/10.4155/fmc-2018-0358>
- Li, Y., Vinyals, O., Dyer, C., Pascanu, R., & Battaglia, P. Learning deep generative models of graphs. arXiv preprint arXiv:180303324.
- Satorras, V. G., Hoogeboom, E., & Welling M. (2021). E(n) equivariant graph neural networks. In *International Conference on Machine Learning: PMLR* (pp. 9323–9332).
- Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:13126114.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).

18. Rezende, D., & Mohamed, S. (2015). Variational inference with normalizing flows. In *International Conference on Machine Learning: PMLR* (pp. 1530–1538).
19. Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. In *Advances in neural information processing systems* (Vol. 33, pp. 6840–6451).
20. Segler, M. H., Kogej, T., Tyrchan, C., & Waller, M. P. (2018). Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Science*, 4(1), 120–131.
21. Bagal, V., Aggarwal, R., Vinod, P., & Priyakumar, U. D. (2021). MolGPT: Molecular generation using a transformer-decoder model. *Journal of Chemical Information and Modeling*, 62(9), 2064–2076.
22. Jin, W., Barzilay, R., & Jaakkola, T. (2018). Junction tree variational autoencoder for molecular graph generation. In *International Conference on Machine Learning: PMLR* (pp. 2323–2332).
23. Jin, W., Barzilay, R., & Jaakkola, T. (2020). Hierarchical generation of molecular graphs using structural motifs. In *International Conference on Machine Learning: PMLR* (pp. 4839–4848).
24. Shi, C., Xu, M., Zhu, Z., Zhang, W., Zhang, M., & Tang, J. (2020). Graphaf: A flow-based autoregressive model for molecular graph generation. arXiv preprint arXiv:200109382.
25. Dinh, L., Sohl-Dickstein, J., & Bengio, S. (2016). Density estimation using real nvp. arXiv preprint arXiv:160508803.
26. Vignac, C., Krawczuk, I., Siraudin, A., Wang, B., Cevher, V., & Frossard, P. (2022). DiGress: Discrete Denoising diffusion for graph generation. arXiv preprint arXiv:220914734.
27. Ragoza, M., Masuda, T., & Koes, D. R. (2022). Generating 3D molecules conditional on receptor binding sites with deep generative models. *Chemical Science*, 13(9), 2701–2713.
28. Luo, Y., & Ji, S. (2022). An autoregressive flow model for 3D molecular geometry generation from scratch. In *International Conference on Learning Representations (ICLR)*.
29. Liu, M., Luo, Y., Uchino, K., Maruhashi, K., & Ji, S. (2022). Generating 3D molecules for target protein binding. arXiv preprint arXiv:220409410.
30. Hoogeboom, E., Satorras, V. G., Vignac, C., & Welling, M. (2022). Equivariant diffusion for molecule generation in 3D. In *International Conference on Machine Learning: PMLR* (pp. 8867–8887).
31. Huang, L., Zhang, H., Xu, T., & Wong, K.-C. (2022). MDM: Molecular diffusion model for 3D molecule generation. arXiv preprint arXiv:220905710.
32. Huang, L. (2023). A dual diffusion model enables 3D binding bioactive molecule generation and lead optimization given target pockets. bioRxiv 2023:2023.01.28.526011.
33. Xu, M., Powers, A., Dror, R., Ermon, S., & Leskovec, J. (2023). Geometric latent diffusion models for 3D molecule generation. arXiv preprint arXiv:230501140.
34. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10684–10695).
35. Zhang, Z., Min, Y., Zheng, S., & Liu, Q. (2023). Molecule generation for target protein binding with structural motifs. In *The Eleventh International Conference on Learning Representations*.
36. Huang, Y., Peng, X., Ma, J., & Zhang, M. (2022). 3Dlinker: An E(3) equivariant variational autoencoder for molecular linker design. arXiv preprint arXiv:220507309.
37. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. In *Advances in neural information processing systems* (Vol. 29).
38. Arjovsky, M., & Bottou, L. (2017). Towards principled methods for training generative adversarial networks. arXiv preprint arXiv:170104862.
39. Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein GAN. arXiv preprint arXiv:170107875.
40. Levina, E., & Bickel, P. (2001). The earth mover's distance is the mallows distance: Some insights from statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision (ICCV 2001)* (pp. 251–256). IEEE.
41. Bai, Q. (2020). Research and development of MolAICal for drug design via deep learning and classical programming. arXiv preprint arXiv:200609747.

42. Kusner, M. J., Paige, B., & Hernández-Lobato, J. M. (2017). Grammar variational autoencoder. In *International Conference on Machine Learning: PMLR* (pp. 1945–1954).
43. Dinh, L., Sohl-Dickstein, J., & Bengio, S. (2016). Density estimation using real NVP. arXiv:160508803.
44. Kingma, D. P., & Dhariwal, P. J. (2018). Glow: Generative flow with invertible 1x1 convolutions. *Adv Neural Inf Process Syst.* 2018;31.
45. Papamakarios G, Pavlakou T, Murray IJAinips. Masked autoregressive flow for density estimation. In *Advances in neural information processing systems* (Vol. 30).
46. Frey, N. C., & Gadepally, V., & Ramsundar, B. (2022). Fastflows: Flow-based models for molecular graph generation. arXiv preprint arXiv:220112419.
47. Madhwawa, K., Ishiguro, K., Nakago, K., & Abe, M. GraphNVP: An invertible flow model for generating molecular graphs. arXiv preprint arXiv:190511600.
48. Welling, M., & The, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International conference on Machine Learning (ICML-11)* (pp. 681–688).
49. Ho, J., Jain, A., & Abbeel, P. J. A. (2020). Denoising diffusion probabilistic models. arXiv:200611239.
50. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., & Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning: PMLR* (pp. 2256–2565).
51. Song, Y., & Ermon, S. J. (2019). Generative modeling by estimating gradients of the data distribution. In *Advances in neural information processing systems* (Vol. 32).
52. Corso, G., Stärk, H., Jing, B., Barzilay, R., & Jaakkola, T. (2022). Diffdock: Diffusion steps, twists, and turns for molecular docking. arXiv preprint arXiv:221001776.
53. Chen, X., Mishra, N., Rohaninejad, M., & Abbeel, P. (2018). Pixelsnail: An improved autoregressive generative model. In *International Conference on Machine Learning: PMLR* (pp. 864–872).
54. LeCun, Y., Chopra, S., Ranzato, M., & Huang, F.-J. (2007). Energy-based models in document recognition and computer vision. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)* (pp. 337–341). IEEE.
55. Xie, J., Zhu, S.-C., & Wu, Y. N. (2019). Learning energy-based spatial-temporal generative convnets for dynamic patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(2), 516–531.
56. Van Den Oord, A., Kalchbrenner, N., & Kavukcuoglu, K. (2016). Pixel recurrent neural networks. In *International Conference on Machine Learning: PMLR* (pp. 1747–1756).
57. Gao, R., Song, Y., Poole, B., & Wu, Y. N., & Kingma, D. P. (2020). Learning energy-based models by diffusion recovery likelihood. arXiv preprint arXiv:201208125.

3D Generative Network



Ran Song, Hao Zhang, and Wei Zhang

Abstract The area of 3D generative network has been developing rapidly due in part to the progresses in generative models and 3D sensing technology. It has a wide range of applications in film and animation, video games, virtual reality, etc. Although many popular 3D generative networks are inspired from the 2D ones, they are significantly different. This is essentially because the representations of 3D data, particularly the non-Euclidean ones, cannot be directly processed by 2D generative networks. In this chapter, we first present an overview of 3D generative networks. Then, we introduce the common representations of 3D data, including Euclidean and non-Euclidean ones. Next, we present the mainstream methods for 3D generative networks categorised subject to the same taxonomy as the 2D generative models. Finally, we discuss the limitations of 3D generative networks and potential future work in this field.

Keywords 3D deep learning · Deep generative network · Generative adversarial network · Diffusion model

1 Introduction

Deep generative networks are a class of generative models based on deep learning, which can generate new data, including images, audio, text, etc. Deep generative networks essentially learn to model the distribution of training data and the generated data are often subject to such a distribution. They have shown great potential in a variety of fields and have become an important topic in artificial intelligence in recent years. The following are common application scenarios of deep generative networks:

R. Song · H. Zhang · W. Zhang (✉)

School of Control Science and Engineering, Shandong University, Jinan, China

e-mail: davidzhang@sdu.edu.cn

R. Song

e-mail: ransong@sdu.edu.cn

H. Zhang

e-mail: haozhangsdu@gmail.com

- Image generation: Deep generative networks can generate various images or to repair and complete damaged images, including natural images, cartoon images, medical images, etc., which leads to applications in video games, virtual reality, and medical science.
- Natural language generation: Deep generative networks can generate natural language texts consistent with human compositions, which facilitates applications such as chatbots, virtual assistants, and content generation.
- Audio generation: Deep generative networks can generate speech highly similar to human speech, which has a range of applications such as text-to-speech transfer, voice assistants, and audio books.
- Data augmentation: Some tasks suffer from small training datasets or data imbalance. Deep generative networks can expand datasets and adjust sample distributions, which typically leads to performance improvement in such tasks.

3D generative networks extend generative networks from 2D to 3D. Most 3D generative networks are inspired by the corresponding 2D versions while taking the specificity of 3D data into account and can be used to generate various types of 3D content, such as models, scenes, animations, and virtual reality experiences. Recently, discussions regarding generative artificial intelligence (AI) in the 3D domain have mainly focused on the simplification of the creative process. Generative AI enables the creation of high-quality 3D models, scenes, and animations within seconds, greatly advancing its applications in creativity and design [1, 21].

The creative process of generative AI typically begins with a prompt, which can be a textual description, a sketch, a photograph, or another 3D model. Various networks are then employed to analyse and process the prompt, generating new 3D content. This content can include models, scenes, animations, or other relevant content related to a specific theme. The models used in generative AI combine multiple algorithms and techniques for representing and processing 3D content. For example, 3D models can be transformed into vector representations, and encoding techniques can be employed to capture their features and structures. It should be noted that these techniques may be influenced by biases, distortions, or incomplete training data when dealing with 3D data. Once developers determine the representation of 3D content, they can apply specific neural networks to generate new 3D content in response to user queries or creative demands [16]. 3D generative networks finds extensive applications in 3D creation, including game development, film production, virtual reality experiences, and so on. It has the capability to automatically generate realistic 3D models, scenes, and animations, significantly improving efficiency and quality in the creative process [3].

The main difference between 2D and 3D generation lies in the data format. Specifically, 2D images can be naturally represented as arrays of pixel values, making them convenient for processing with neural networks. In contrast, there are multiple representation methods for 3D instances, such as point clouds [13, 14], meshes [18, 20], voxel grids [8, 22], multi-view images [23], and implicit neural representations [10]. Each representation method has its own advantages and limitations. For example, meshes compactly represent 3D shapes but are challenging to analyse and generate

through neural networks due to the irregularity of their data structure. On the other hand, voxel grids have regular positioning in 3D space and work well with standard convolutional neural networks. However, voxel grids consume significant memory and struggle to represent high-resolution 3D scenes. Selecting the appropriate 3D representation form is crucial based on different applications and requirements. For instance, point cloud representations may be more suitable for handling real-world 3D objects or scenes as they directly capture the geometric structure and details of objects. In virtual reality or game development, voxel grids may be more commonly used as they provide better control over object volume and surface.

As generative AI continues to evolve in the 3D domain, researchers and engineers are constantly exploring more advanced 3D representation and generation techniques. For example, implicit neural representations leverage the mapping relationship from a low-dimensional latent space to 3D scenes, enabling efficient generation of high-quality 3D content. This approach can overcome limitations of traditional representation methods and reduce storage and computational resource requirements. Overall, selecting the appropriate 3D representation form is crucial for the successful application of generative networks in the 3D domain. Different representation forms have their own advantages and applicability, while researchers and engineers strive for continuous improvement and innovation to enhance the quality and efficiency of 3D content generation.

In the following, we shall first introduce the representation of 3D data, and then present the mainstream methods for 3D deep generative networks, and finally discuss the limitations of 3D generative networks.

2 3D Data Representation

3D data can be represented in different ways, divided into Euclidean and non-Euclidean representations. Euclidean 3D data has an underlying grid structure, which allows for global parameterisation and a common coordinate system. These characteristics make it straightforward to extend existing 2D deep learning frameworks to 3D data, where convolution operations are similar to those in 2D. On the other hand, non-Euclidean 3D data lacks a grid-like array structure and global parameterisation. Therefore, extending classical deep learning techniques to such representations is a challenging task. In real-life scenarios, studying deep learning techniques in the non-Euclidean domain holds significant importance. This is referred to as geometric deep learning.

2.1 Euclidean Data

Euclidean data preserves the attributes of a grid structure and possesses global parameterisation and a common coordinate system. The primary representations of this type of 3D data include voxel grids and multi-view images.

Voxel Grids. Voxel grids can be used to represent individual samples or data points on a three-dimensional grid with regular intervals, similar to pixels in a two-dimensional space. Data points can contain single data, such as opacity, or multiple data, such as color and opacity. Voxel grids can also store high-dimensional feature vectors within data points, such as geometric occupancy, volume density, or signed distance values. Voxel grids represent only a point on the grid, not a volume; the spatial relationship between voxels is not explicitly represented in voxel-based datasets. Depending on the data type and the intended use of the dataset, the missing information can be reconstructed and/or approximated, for example, through interpolation.

Voxel representations are simple, with a clear spatial structure and high scalability, making them easily applicable to convolutional neural networks. However, they have lower efficiency as they represent both occupied and unoccupied parts of the scene, leading to significant storage requirements for unnecessary data. This makes voxel grids less suitable for representing high-resolution data. Voxel grids have numerous applications in rendering tasks. Early methods stored high-dimensional feature vectors within voxels to encode the geometric shape and appearance of scenes, often referred to as feature volumes, which can be interpreted as color images using projection and 2D cellular neural networks. This also includes volume imaging in medicine and terrain representation in games and simulations.

Multi-View Images. With the development of computer vision technology and significant improvements in computational power, coupled with the latest advancements in digital cameras, it is now easy to capture a large number of high-resolution images. For many applications, there is an urgent need to extract 3D structures from these images, such as 3D reconstruction. Multi-view image datasets are aggregations of multiple images, each representing an object or scene from different perspectives (e.g. front, side, and top), combined to form a multi-view image dataset. The availability of multi-view image datasets is a significant advantage due to the time-consuming process of collecting 3D data from the real world and the reliance of deep learning paradigms on large amounts of data for training. However, multi-view images cannot be strictly defined as 3D model data, but they serve as a bridge between 2D and 3D visualisation. Recently, NeRF has emerged as a novel approach for 3D reconstruction, particularly well-suited for the massive data requirements of generalised NeRF methods based on learning. It can also be applicable to multi-view stereo and view-consistency image understanding tasks.

2.2 Non-euclidean Data

The second type of 3D data representation is non-Euclidean data. This type of data lacks global parameterisation or a common coordinate system, making it challenging to extend 2D deep learning paradigms. Significant efforts have been made to study and apply deep learning techniques in this data representation, known as geometric deep learning. The primary types of non-Euclidean data are point clouds, 3D meshes, and implicit data.

Meshes. 3D meshes are one of the most popular representations of 3D shapes. A 3D mesh structure consists of a collection of polygons called faces, which are described based on a set of vertices that represent the existence of coordinates in 3D space. These vertices are associated with a connectivity list that describes how these vertices are connected to each other. Meshes only model the surface of a scene, making them more compact. Meshes provide connectivity of surface points for modeling point relationships. Due to these advantages, polygonal meshes have been widely used in traditional computer graphics applications such as geometry processing, animation, and rendering. However, at a global level, meshes are non-Euclidean data, where the local geometry of the mesh can be represented as a subset of Euclidean space, without the known properties of Euclidean space such as translational invariance, vector space operations, and global parameterisation systems. Therefore, deep learning on 3D meshes is a challenging task. However, with the development of graph neural networks, meshes can be seen as graphs. For example, MeshCNN [5] is specifically designed for convolutional and pooling layers on mesh edges, extracting edge features for shape analysis. 3D meshes are essential in various fields and industries such as architecture and construction, furniture and home living, gaming and entertainment, product design, medical and life sciences, among others.

Point Clouds. With the trend of affordable and user-friendly point cloud acquisition devices, point clouds have been widely used in fields such as modeling and rendering, augmented reality, autonomous driving cars, and so on. Point clouds are an unordered collection of discrete samples of 3D shapes in 3D space. Traditionally, point clouds are considered non-Euclidean data since point cloud data is globally unstructured. However, point clouds can also be realized as a set of locally parameterised small subsets of Euclidean space. The definition of the point cloud structure depends on whether the global or local structure of objects is considered. Since most applications strive to capture the global characteristics of objects to perform complex tasks, traditional point clouds are considered non-Euclidean data. Point clouds are directly output from depth sensors, making them highly popular in 3D scene understanding tasks.

Despite the ease of acquiring point clouds, their irregularity makes them challenging to process with traditional 2D neural networks. Numerous geometric deep learning methods have been proposed to effectively analyze 3D point clouds, such as PointNet [13], which is a deep learning network structure based on raw point cloud data. It can directly use raw point cloud data as input and aggregate the input point cloud using a set of sparse key points, effectively handling data with robustness

against small perturbations in the input and achieving good performance in shape classification, part segmentation, scene segmentation, and other tasks. 3D point cloud technology has applications in various fields and industries, including architecture, engineering, civil building design, geological surveys, computer vision, agriculture, spatial information, autonomous driving, and can provide more accurate modeling and analysis, as well as more precise data processing.

Neural Fields. A neural field is a domain in 3D space representing a scene or object that is fully or partially parameterised by the parameters of a neural network and is represented by the attributes of the neural network at each point in 3D space. Neural fields have continuous representational capacity and can represent 3D scenes or objects at any resolution and with unknown or complex topology. Additionally, compared to the aforementioned representations, neural fields only require storing the parameters of the neural network, resulting in lower memory consumption compared to other representations.

3 3D Generative Networks

2D generative networks can be categorised into several groups, including energy-based models (EBM) [19], variational autoencoders (VAE) [6], generative adversarial networks (GAN) [4], autoregressive models (ARM) [11], normalised flows (NF) [7], and diffusion models (DM) [15]. Since many 3D generative networks extends from such approaches, we follow such a taxonomy to introduce 3D generative networks.

3.1 Energy-Based Models

EBM is based on the observation that any probability density function $p(x)$ for $x \in \mathbb{R}^D$ can be expressed in terms of an energy function $E(x) : \mathbb{R}^D \rightarrow \mathbb{R}$:

$$p(x) = \frac{e^{-E(x)}}{\int_{\tilde{x} \in X} e^{-E(\tilde{x})}}. \quad (1)$$

The energy function is typically defined as a neural network that produces a scalar output, which represents the energy value of the input. It is trained to assign low energy values to the instances in the training data and high energy values to instances that are unlikely to be sampled from the true data distribution. In traditional maximum likelihood estimation, we seek to maximize the log-likelihood of the model parameters given the data. However, this is often intractable for EBM, as the denominator in Eq. (1) is difficult to compute exactly. Therefore, contrastive divergence is widely used to train EBM. This algorithm is a Markov Chain Monte Carlo method that runs a short Gibbs sampling chain to generate negative samples,

which are then used to update the model parameters so that the discrepancy between the energy distribution of the model and the true data distribution can be reduced.

EBM uses energy functions to explicitly model the probability distribution of 3D data. They are based on a fundamental idea that any probability function can be transformed from an energy function by normalising its volume:

$$p(x) = \frac{\exp(-E_\theta(x))}{\int_x \exp(-E_\theta(x))} \quad (2)$$

where $-E_\theta(x)$ represents the energy function. Clearly, 3D data points with high probability have lower energy, while data points with low probability have higher energy. However, due to computational reasons, computing the normalisation constant $\int_x \exp(-E_\theta(x))$ for $p(x)$ is difficult for high-dimensional 3D data. To alleviate such an optimisation challenge, the idea of contrastive divergence is introduced by comparing the probability gradient of $p(x)$ and the data randomly sampled from the energy distribution $q_\theta(x)$. Contrastive divergence aims to minimise the difference between the parameters θ of the energy function and the 3D data distribution. This is achieved by maximising the difference between the probability gradient on $p(x)$ and the random sampling on $q_\theta(x)$:

$$\nabla_\theta \mathbb{E}_{x \sim q_\theta} (-\log(p(x))) = \mathbb{E}_{x \sim p}(E_\theta(x)) - \mathbb{E}_{x \sim q_\theta}(E_\theta(x)). \quad (3)$$

In this way, we can approximate the latent features of the 3D data distribution during the energy function optimisation. Contrastive divergence provides a more stable approach for optimising energy models, particularly suitable for handling high-dimensional 3D data since it does not require the computation of the normalisation constant. By leveraging the advantages of contrastive divergence and energy functions, EBM can effectively learn and generate high-quality 3D data samples and estimate 3D data distribution. The design and adjustment of contrastive divergence remain an active research area, and researchers are continuously striving to improve the performance and stability of EBM in the 3D domain.

In Eq. (3), the energy distribution $q_\theta(x)$ is approximated through Markov Chain Monte Carlo (MCMC) processes. In MCMC, we use Markov chains to sample from the probability distribution $q_\theta(x)$ corresponding to the energy function $E_\theta(x)$. By iteratively applying state transition operations, we can generate a series of samples starting from an initial state that gradually converges to the target energy distribution. The MCMC process is based on the Markov property, where the current state is only dependent on the previous state. By defining appropriate state transition operations, such as the Metropolis-Hastings algorithm or Gibbs sampling, we can achieve transitions from the current state to the next state. By repeating this transition process, we can explore the sample space in the Markov chain and eventually obtain a set of samples from the energy distribution $q_\theta(x)$. Using MCMC to approximate the energy distribution $q_\theta(x)$ allows EBM to generate high-quality 3D data samples from the latent space. The MCMC process helps us traverse the latent space and capture the characteristics of the data distribution. However, MCMC computations are typically

expensive, especially in high-dimensional 3D data spaces. Therefore, researchers have been working on improving MCMC algorithms and developing more efficient sampling methods to enhance the performance and scalability of EBM in the 3D domain.

3.2 Variational Autoencoders

VAE is based on the idea of variational inference, where a probabilistic model is approximated by a simpler distribution in order to make inference tractable. In the case of VAE, the model is a generative model that generates data by sampling from a latent variable, which is assumed to follow a simple prior distribution (e.g., a standard normal distribution). The goal of VAE is to learn an approximation of the posterior distribution over the latent variables, given a set of observed data points. To achieve this, VAE uses two neural networks: an encoder network that maps the input data to a distribution over the latent variables, and a decoder network that maps a sample from the latent space to a reconstructed output. During training, VAE optimises two objectives: the reconstruction loss, which measures the difference between the input data and the output of the decoder network, and the Kullback-Leibler (KL) divergence between the encoder distribution and the prior distribution, which encourages the learned latent distribution to be similar to the prior. By optimising the two objectives, VAE learns a compressed representation of the data that can be used to generate new samples by sampling from the learned latent distribution. VAE has been used in a wide range of applications, including image generation, natural language processing, and anomaly detection.

3D VAE is a deep generative network used for modelling 3D data distributions. Unlike 2D images, the representation and processing of 3D data are more challenging. 3D VAE addresses this problem by introducing latent variables z and using neural networks to parameterize the 3D data distribution $x \sim p_\theta$. However, compared to 2D VAE, 3D VAE faces more complex modelling and inference tasks. The core idea of 3D VAE is to approximate the intractable posterior distribution by an encoder network $q_\phi(z|x)$. The encoder network maps the input 3D data x to latent variables z in the latent space. This process utilises a feedforward neural network structure and minimises the KL-divergence to optimise the difference between the encoder network and the true posterior distribution $p_\theta(z|x)$. Consequently, we can generate new 3D samples by sampling $z \sim q_\theta(z|x)$ from the encoder network:

$$\begin{aligned} D_{KL}(q_\theta(z|x) || p_\theta(z|x)) &= \log(p_\theta(x)) + D_{KL}(q_\theta(z|x) || p_\theta(z)) \\ &\quad - \mathbb{E}_{z \sim q_\theta(z|x)} \log(p_\theta(z|x)) \end{aligned} \quad (4)$$

where $D_{KL}(\cdot || \cdot)$ denotes the KL divergence between two variables. The log-likelihood of $p_{\theta(x)}$ can be rewritten as:

$$\begin{aligned}
\log(p_\theta(x)) &= D_{KL}(q_\theta(z|x) || p_\theta(z|x)) - D_{KL}(q_\theta(z|x) || p_\theta(z)) \\
&\quad + \mathbb{E}_{z \sim q_\theta(z|x)} \log(p_\theta(z|x)) \\
&\geq -D_{KL}(q_\theta(z|x) || p_\theta(z)) + \mathbb{E}_{z \sim q_\theta(z|x)} \log(p_\theta(z|x))
\end{aligned} \tag{5}$$

where the term $D_{KL}(q_\theta(z|x) || p_\theta(z))$ can be eliminated since the KL divergence is always non-negative. The above equation defines the loss function of a VAE, expressed as:

$$\mathcal{L}_{VAE} = -D_{KL}(q_\theta(z|x) || p_\theta(z)) + \mathbb{E}_{z \sim q_\theta(z|x)} \log(p_\theta(z|x)). \tag{6}$$

In addition, the training process of 3D VAE typically involves a reconstruction-based loss function that encourages the model to learn to reconstruct input data into high-quality 3D samples. The stability of this training process benefits from the feed-forward structure of the encoder network and effective inference methods. However, similar to 2D VAE, 3D VAE may also face the problem of posterior collapse, where the learned latent space cannot fully reconstruct the given 3D data. To overcome these challenges, researchers are continuously improving the architecture and training methods of 3D VAE. For example, introducing more complex neural network structures, using loss functions better suited for 3D data, and improved reparameterisation techniques are important directions for improving the performance and generation quality of 3D VAE. These efforts aim to enable 3D VAE to better handle and generate realistic 3D data, providing powerful generative models for fields such as 3D graphics, computer-aided design, and virtual reality.

3.3 Generative Adversarial Networks

3D GAN is highly favoured in the field of 3D data generation due to their excellent data synthesis capabilities. Generally, a 3D GAN consists of two independent networks: a generator $G(\cdot)$ and a discriminator $D(\cdot)$. The generator $G(\cdot)$ takes latent codes sampled from a prior distribution $z \sim p_z$ as input and generates 3D data samples, while the discriminator $D(\cdot)$ aims to accurately distinguish real 3D data $x \sim p_x$ from the synthesized 3D data $G(z)$ generated by the generator $G(\cdot)$. During the training process, the generator $G(\cdot)$ strives to synthesize realistic 3D data samples to fool the discriminator $D(\cdot)$, making it unable to accurately distinguish between synthetic and real samples. At the same time, the discriminator $D(\cdot)$ is trained to correctly classify generated samples as fake and real samples as real. These two networks compete with each other, forming a minimax game (zero-sum game), and their performance is improved through joint optimization. The loss functions for training $G(\cdot)$ and $D(\cdot)$ can be expressed as

$$\begin{aligned}\mathcal{L}_D &= -\mathbb{E}_{x \sim p_x} [\log(D(x))] - \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))] , \\ \mathcal{L}_G &= -\mathbb{E}_{z \sim p_z} [\log(D(G(z)))].\end{aligned}\quad (7)$$

With the rapid development of deep learning, the network structure of 3D GAN has also evolved, such as 3D-GAN [21], Pi-GAN [3], LiftedGAN [17] and EGA 3D-GAN[2]. Despite the significant achievements of 3D GAN in 3D data synthesis, there are still challenges. Due to the high dimensionality and complexity of 3D data, training 3D GAN networks often requires substantial computational resources and time. Additionally, fine modelling of 3D shapes and structures is also challenging. Researchers are devoted to improving training algorithms, network architectures, and evaluation metrics for 3D GAN to further advance the field of 3D data generation and provide more powerful tools for applications such as virtual reality, game development, and medical image processing.

3.4 Autoregressive Models

ARM is a type of generative model that generates data by modelling the conditional probability distribution of the next value in a sequence given the previous values in that sequence. ARM is often used in the context of deep learning, where neural networks are used to model the conditional probabilities. The following are several common ARMs:

- (1) Language model: The language model predicts the next word based on the words generated previously, and repeats this process until a complete sentence is generated. Common language models include the generative pre-trained transformer (GPT) series, bidirectional encoder representations from transformers (BERT), embeddings from language model (ELMo), and so on.
- (2) Image generation model: The image generation model typically treats the image as a sequence of pixels and generates each pixel in sequence. Common image generation models include PixelCNN, PixelRNN, and so on.
- (3) Audio generation model: The audio generation model typically treats audio as a time-domain signal and generates each sample sequentially in time steps. Common audio generation models include WaveNet [12], SampleRNN [9], and so on.

Overall, ARM is a powerful type of generative model that can generate 2D or 3D data with certain structures and patterns, and are widely used in various fields.

3.5 Normalizing Flows

NF is a type of generative model used in machine learning to learn the underlying probability distribution of a dataset. Unlike other generative models, such as GAN or

VAE, which generate samples through a deterministic function, NF generates samples through a sequence of invertible transformations, also known as “flows”. The goal of NF is to transform a simple distribution, such as a standard normal distribution, into a more complex distribution that matches the distribution of the dataset. The transformations are designed in such a way that the resulting distribution can be easily sampled from and the likelihood can be easily computed. In a normalising flow, each transformation is chosen to be an invertible function with tractable Jacobian determinant. By chaining together multiple invertible functions, the flow can create a complex non-linear transformation that can approximate any continuous probability distribution. During the training process, the model learns to adjust the parameters of the transformations to maximise the likelihood of the dataset.

NF has been widely used in 3D data generation and involved introducing a set of invertible transformation functions to address the challenge of computationally intractable likelihood functions in parameterised models. In 3D data generation, NF starts with a known simple distribution, such as a multidimensional Gaussian distribution, and gradually transforms it through a series of invertible functions f_1, f_2, \dots, f_N to obtain the desired output probability distribution:

$$z_i = f_{i-1}(z_{i-1}). \quad (8)$$

The entire chain composed of z_i is called an NF. Each invertible function f maps the input variable to the output variable, and its inverse function f^{-1} is also invertible. This allows us to compute the gradient of the transformed probability density function through forward and backward propagation. Due to the invertibility of f_i , the probability density function of the new variable z_i can be easily estimated from the previous step z_{i-1} :

$$\begin{aligned} p(z_i) &= p(z_{i-1}) \left| \frac{df_i}{dz_{i-1}} \right|^{-1}, \\ \log p(z_i) &= \log p(z_{i-1}) - \log \left| \frac{df_i}{dz_{i-1}} \right|. \end{aligned} \quad (9)$$

According to the chain rule, the density of the final output z_N after N transformations can be obtained by the following equation:

$$\log p(z_N) = \log p(z_0) - \sum_{i=1}^N \log \left| \frac{df_i}{dz_{i-1}} \right|. \quad (10)$$

The key goal of NF in 3D data generation is to construct a complex transformation through these invertible transformation functions, such that the final output probability distribution can approximate the true distribution of 3D data. By training the parameters of the invertible functions, NF can gradually adjust the transformation to make the generated 3D samples closer to the distribution of real data. This approach allows us to compute the gradient of the probability density function during the

training process, enabling the use of optimisation methods such as gradient descent to update the model parameters. However, NF often faces a trade-off between the capacity and efficiency of parameterised models. Larger model capacity can better approximate complex data distributions but increases computational complexity and training difficulty. Conversely, smaller model capacity may fail to capture subtle variations and complex structures in the data. Therefore, balancing the capacity and efficiency of parameterised models is an important consideration in the design and training of NF. Researchers typically address this issue by improving the structure of transformation functions and introducing regularisation techniques. For example, deep neural network structures can be used to enhance the non-linear expressive power of the model, while regularisation methods can control model complexity to avoid overfitting. By striking a balance between model capacity and efficiency, normalisation flows can play a crucial role in 3D data generation.

NF has significant applications in 3D data generation, providing more accurate likelihood function estimation and improving the training and generation performance of models. By utilising invertible transformation functions, NF can achieve high-quality 3D sample generation and better optimise the objective function during model training. This makes NF a powerful tool for handling complex 3D data distributions and generating realistic 3D samples.

3.6 Diffusion Models

DM represents a recent emerging topic in computer vision, demonstrating remarkable results in the area of generative modelling. A diffusion model is a deep generative model based on two stages, a forward diffusion stage and a reverse diffusion stage. In the forward diffusion stage, the input data are gradually perturbed over several steps by adding Gaussian noise. In the reverse stage, a model is tasked at recovering the original input data by learning to gradually reverse the diffusion process, step by step. DM is widely appreciated for the quality and diversity of the generated samples, despite their known computational burdens, i.e. low speeds due to the high number of steps involved during sampling. DM also allows the control over specific features of the generated data. For example, it is possible to condition the model on certain attributes, such as the presence of a particular object or scene, and generate images that contain those attributes.

DM is parameterised a Markov chain model that models data by gradually adding noise to the input data x_0 , where the noise schedule $\beta_1 : T$ represents the time steps. In theory, as T approaches infinity, x_T tends to a normal Gaussian distribution \mathcal{N} , expressed as

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t I),$$

$$q(x_{1:t}|x_0) = \prod_{t=1}^T q(x_t|x_{t-1}). \quad (11)$$

where I denotes the identity matrix. To learn the reverse process of the diffusion, we establish a parameterised model p_θ to model the conditional transition probability $q(x_{t-1}|x_t)$ from noise to data. To optimise this model, we use the evidence lower bound (ELBO) method. Due to the complexity of handling long Markov chains, DM can generate high-quality data samples and has relatively stable training processes. However, DM has a higher training cost and slower sampling process compared to GAN and VAE. Nonetheless, DM has the potential for data synthesis and generating high-quality samples, especially when dealing with complex distributions of 3D data. Researchers are actively working on improving the efficiency and sampling speed of DM to enhance the practical utility.

4 Discussion

Despite their potential, the new generation of 3D generative networks present a Pandora’s box of challenges related to accuracy, trustworthiness, bias, hallucination, and plagiarism—ethical issues that are likely to take years to untangle. The convincingly realistic nature of generative AI content introduces a fresh set of AI risks in the 3D domain. It becomes increasingly difficult to discern AI-generated content, and more importantly, to detect when things go awry. This poses a significant problem when we rely on generative AI outputs for tasks such as coding or providing medical guidance. Many outcomes produced by generative AI lack transparency, making it challenging to ascertain if they potentially infringe upon copyrights or if there are underlying issues with the original sources they draw upon. Without an understanding of how a 3D generative network arrives at its conclusions, we cannot effectively reason about the potential for errors.

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., & Guibas, L. (2018). Learning representations and generative models for 3d point clouds. In *ICML* (pp. 40–49).
2. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L.J., Tremblay, J., & Khamis, S., et al. (2022). Efficient geometry-aware 3D generative adversarial networks. *CVPR* pp. 16123–16133.
3. Chan, E.R., Monteiro, M., Kellnhofer, P., Wu, J., & Wetzstein, G. (2021). Pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis. In *CVPR* (pp. 5799–5809).
4. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *NeurIPS* (pp. 2672–2680).

5. Hanocka, R., Hertz, A., Fish, N., Giryes, R., Fleishman, S., & Cohen-Or, D. (2019). MeshCNN: a network with an edge. *ACM Transactions on Graphics*, 38(4), 1–12.
6. Kingma, D.P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint [arXiv:1312.6114](https://arxiv.org/abs/1312.6114).
7. Kobyzhev, I., Prince, S. J., & Brubaker, M. A. (2020). Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11), 3964–3979.
8. Maturana, D., & Scherer, S. (2015). VoxNet: A 3D convolutional neural network for real-time object recognition. In *IROS* (pp. 922–928).
9. Mehri, S., Kumar, K., Gulrajani, I., Kumar, R., Jain, S., Sotelo, J., Courville, A., & Bengio, Y. (2016). Samplernn: An unconditional end-to-end neural audio generation model. arXiv preprint [arXiv:1612.07837](https://arxiv.org/abs/1612.07837).
10. Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2021). NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1), 99–106.
11. Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., & Graves, A., et al. (2016). Conditional image generation with pixelcnn decoders. In *NeurIPS*.
12. Oord, A.v.d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., & Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. arXiv preprint [arXiv:1609.03499](https://arxiv.org/abs/1609.03499).
13. Qi, C.R., Su, H., Mo, K., & Guibas, L.J.: PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR* (pp. 652–660).
14. Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS* (pp. 5099–5108).
15. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *CVPR* (pp. 10684–10695).
16. Schwarz, K., Liao, Y., Niemeyer, M., & Geiger, A. (2020). GRAF: Generative radiance fields for 3d-aware image synthesis. In *NeurIPS* (pp. 20154–20166).
17. Shi, Y., Aggarwal, D., & Jain, A. K.: Lifting 2D styleGAN for 3D-aware face generation. In *CVPR* (pp. 6258–6266)
18. Sinha, A., Unmesh, A., Huang, Q., & Ramani, K. (2017). SurfNet: Generating 3D shape surfaces using deep residual networks. In *CVPR*.
19. Song, Y., & Kingma, D. P. (2021). How to train your energy-based models. arXiv preprint [arXiv:2101.03288](https://arxiv.org/abs/2101.03288).
20. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., & Jiang, Y. G.: Pixel2Mesh: Generating 3D mesh models from single RGB images. In *ECCV* (pp. 52–67).
21. Wu, J., Zhang, C., Xue, T., Freeman, B., & Tenenbaum, J. (2016). Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *NeurIPS*.
22. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., & Xiao, J. 3D shapenets: A deep representation for volumetric shapes. In *Proceedings on CVPR* (pp. 1912–1920).
23. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., & Snavely, N. (2018). Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics*, 37(4), 1–12.

The Economics of Generative AI



Stanislav Ivanov

Abstract The chapter focuses on the economic aspects of generative AI. It looks at the cost–benefit analysis of generative AI implementation in a company, delves into the automatability of tasks by generative AI, evaluates its substitution, enhancement (augmentation) and transformational effects, discusses its economic limitations and its micro-and macroeconomic implications. The analysis will be helpful to managers and owners who consider the adoption of generative AI by their organisations.

Keywords Generative AI · Artificial intelligence · Economics · Cost–Benefit analysis · Task automatability · Substitution effect · Enhancement effect · Transformational effect

1 Introduction

Generative artificial intelligence (GenAI) is AI that is used to create content as output—text, audio, pictures, code, video, etc. [1]. Although it has been around for quite some time, it took the world by storm after ChatGPT was released in November 2022. It has many useful applications that make it attractive to companies—integration into chatbots where it improves the customer-chatbot interactions and makes them more human-like, preparation of draft documents such as contracts, check lists, press releases; analysis and summary of customer reviews; generating ideas for new recipes, menu, or thematic dish names; creating pictures; creating video presentations with digital humans (avatars); drafting software code; translate or rephrase text; generate academic text; automating tasks in education, journalism, and countless other applications [1–8]. Companies may utilise it to cut costs, improve efficiency and productivity by automating (some) cognitive tasks, inspiration/generating ideas

S. Ivanov (✉)

Varna University of Management, 13A Oborishte Str, 9000 Varna, Bulgaria

e-mail: info@zangador.institute; stanislav.ivanov@vumk.eu

URL: <http://www.stanislavivanov.com>

Zangador Research Institute, 9010 Varna, Bulgaria

for innovations, improving service quality, and other economic reasons. They may also embark the generative AI boat just because it is fashionable and competitors use it (the so called ‘mimetic pressure’ from the Institutional theory of [9]. Recently, [10] found that exposure to ChatGPT influences positively the value of a firm. Regardless of the reasons, the successful implementation of generative AI in a company needs to follow the basic economic logic—adopt it when it improves the economic performance of the company (regardless of how it is measured—profitability, sales, market share, market capitalization, etc.), consider the benefits but also the costs of using (or not using) generative AI, and pay attention to its broader micro- and macroeconomic implications that go beyond the boundaries of the firm.

This chapter focuses on the economic aspects of generative AI. It looks at the cost-benefit analysis of generative AI implementation in a company, delves into the automatability of tasks by generative AI, evaluates its substitution, enhancement (augmentation) effects, discusses its economic limitations, and its micro-and macroeconomic implications.

2 Cost–Benefit Analysis

Generative AI will have significant economic implications for companies and organisations and their evaluation requires that managers implement cost-benefit analysis before adopting the new technology [11]. The directions of the analysis need to go beyond the pure financial costs and benefits and incorporate the non-financial ones as well, i.e. those related to marketing, human resources, operations, and strategic management.

2.1 Benefits

- *Productivity gains*—generative AI allows employees to generate high-quality content quickly. Generative AI applications can rephrase or translate texts, create draft versions of photos, contracts, job descriptions, check lists, programming code, presentation templates, or other content that can be edited by human employees. In that way, AI saves time and efforts to employees and increases their productivity.
- *Cost efficiency*—adopting generative AI to automate tasks can bring about significant cost reductions. For example, it can minimize the expenses for acquiring stock photos, creating and translating written content, or employing human actors for advertisements. Integrating it into a chatbot enhances its efficiency in interactions with customers, reducing the need for human intervention and subsequently, labour costs. Productivity gains drive the costs down as well. However, it is necessary to acknowledge that the current generative AI technology is not flawless. Its

output might be wrong, irrelevant, or insufficiently specific and may require additional editing by a human. Thus, employees might need to produce more content than required to ensure an adequate amount of usable content, potentially incurring additional costs. Nevertheless, the initial output from the AI can serve as a draft, saving both time and money for companies.

- *Decent work*—generative AI can help companies create a decent work environment for their employees that implement cognitive tasks. By generating drafts of texts, code, pictures and videos, generative AI automates many tasks which frees time to employees to deal with more creative tasks.
- *Stimulating innovations*—generative AI gives an additional technological arm to companies which they can use to create ideas for new products and services, redesign of business models, formulate brand names, slogans and logos, identify new market segments, etc. These can be further refined and upgraded by human employees who may provide even more ideas for innovation inspired by the AI's suggestions.
- *Improved product/service quality*—the improved customer experience with LLM-based chatbots, the relevance of medical/financial advice they give, or the higher quality of the pictures that AI generates are a few examples of the ways generative AI can contribute to increasing the product/service quality.
- *Competitiveness*—the preceding benefits hint on the ways generative AI can improve the competitiveness of companies: by decreasing a company's costs and prices, by improving the product/service quality, and by shortening the time necessary to fulfil a purchase order (e.g. delivering the legalised translation of a document in a couple of hours instead of a day).
- *Revenue generation*—generative AI can directly and indirectly stimulate revenue. An LLM-based chatbot can provide more personalized and relevant information to prospective guests and more human-like interactions with them compared to other chatbots which could result in more bookings and, thus, revenue [2]. Furthermore, the quick generation of high-quality and diverse content allows managers to create and test numerous versions of a marketing campaign that could appeal to a wider customer audience, indirectly increasing the company's revenue. The ideas generated by AI regarding new products, branding, promotion strategies, and business models can also inspire managers to innovate, enhancing competitiveness and revenue growth indirectly.

2.2 Costs

- *Technology-related costs*—generative AI is not free; someone needs to pay for it in one way or another. Although many generative AI applications offer some limited free service, most of the functions are accessible against a monthly fee. The latter can range from tens to thousands of euro and depends on the company, the availability of competing products, the technical functionalities of the AI application the subscriber wants to use, the number of users (for corporate customers),

etc. The monthly fee creates a steady cash inflow for the software companies but at the same time it is a fixed cost for their corporate customers. The monthly basis of these costs make them directly comparable to the labour costs thus facilitating the evaluation of the potential substitution of labour for AI. Additional technology-related costs include the initial investment needed to train the AI with company-specific data to provide company-specific responses.

- *Dependence on AI*—one of the major issues related to the implementation of a new technological solution is the potential vendor lock-in effect—high switching costs and dependence on the supplier due to incompatibility of spare parts or potential loss of data [12]. The generative AI does not cause a vendor lock-in effect unlike other technological solutions because it uses the ‘Technology-as-a-service’ model, i.e. a monthly subscription to access the website/app. However, generative AI may cause another type of dependence—not on the vendor but on AI itself. The ease of generating draft content (text, video, code, pictures) means that employees may rely too much on AI to create the initial drafts of these, hence, some loss of creativity and innovativeness skills of employees might be observed, while the generated output might be too standardised due to the inherent limitations of the models of the computer algorithms of the respective AI application.
- *Ethical issues* arise with all AI technologies [13]. In the context of generative AI, they relate to issues such as biases in the training datasets, biases in the computer models, or potential generation of insulting, socially inappropriate or fake content. Embedding limitations in the computer models regarding the acceptability in their output may help mitigate some of the ethical issues but raises questions about the limitations themselves—why these limitations and not others? AI ethics is a vast field and there are many issues that are yet to be solved [14, 15].
- *Generative AI as a threat*—people may perceive generative AI as a threat to their jobs (replacing them), to social norms and culture (creating socially unacceptable content or fake news), or even as an existential threat to humans as a species (potentially become sentient). Regardless of the reasons, people may reject the use of generative AI and boycott companies that use it.
- *Costs of non-adoption (missed benefits)*—non-adoption of generative AI might result in indirect economic costs due to a potential decrease in the company’s competitiveness.

3 Automatability of Tasks

A key issue in the economics of generative AI is the automatability of tasks that need to be performed in a company. Automatability refers to how easy it is a task to be performed by technology rather than by a human employee and it depends on task’s characteristics: nature, complexity, frequency, and standardization.

- **Nature of the task**—physical versus cognitive tasks. Automating physical tasks (e.g. moving items, cleaning) and cognitive tasks (e.g. calculations or playing chess) requires different technologies. Physical tasks require mobility and are usually automated through some type of industrial or service robots, or other devices while cognitive tasks may be automated through software/intelligent automation [16] that does not need to be embedded in a device. Physical tasks are more complex than cognitive tasks, as they necessitate the technology, such as a mobile service robot, to gather, analyze, and respond to real-time data from its surroundings. Hence, these physical tasks inherently include cognitive elements as well. Simultaneously, tasks demanding high emotional intelligence like handling customer complaints are more challenging to automate due to the limited emotional intelligence capabilities of automation technologies. Generative AI is used to automate cognitive tasks.
- **Task complexity**—it is easier to automate simple and routine tasks (e.g., inputting customer data in a database) than complex tasks that require high level of critical thinking and creativity (e.g., evaluating the relevance of a previous court decision to a particular legal case).
- **Task frequency**—repetitive tasks create economies of scale for automation technologies because a technological solution automates many tasks, often across job positions, and this makes its automation economically feasible for the company. One-off or rarely performed tasks are less worthy to automate.
- **Task standardization**—usually tasks that have a well-defined and consistent procedure are automatable.

In this framework, simple, routine, repetitive and standard cognitive tasks are highly automatable, while complex, one-off/rare, and diverse physical tasks are less likely to be automated. However, advances in generative AI blurred this distinction among tasks. Now AI can generate pictures, poems, and jokes—all highly creative tasks that were previously considered reserved to humans. In companies, AI can generate draft labour contract or offers to customers, which managers can edit. It can develop a thematic menu in a restaurant, an itinerary for a city break trip to a destination, or a recipe [2, 4, 5]. Therefore, generative AI expands the scope of automatability of tasks compared to kiosks, robots, robotic process automation, and other technologies. In fact, complex, creative and diverse cognitive tasks (e.g. writing a science fiction novel) become automatable as well. This raises the requirements towards the human employees who perform such tasks and they need to improve their skills because their job positions are not automation-proof. They would need to learn how to collaborate with generative AI and use it effectively, efficiently and creatively in their work. Learning how to operationalize complex tasks into a set of simpler tasks and how to develop appropriate prompts to generate relevant content (text, pictures, tables, code) in AI applications becomes of crucial importance, because the work of the future will be largely based on ‘human-AI collaboration’ (see also [17]).

4 Enhancement, Substitution and Transformational Effects of Generative AI

Generative AI has three distinct effects on labour (enhancement, substitution, and transformational effects) based on its impacts on the tasks that human employees need to perform after the implementation of a generative AI application in the company (see also [18, 19]). Through the *substitution effect*, generative AI automates most of the tasks that were previously implemented by human employees so that they are left with too few tasks to do and sustaining their jobs becomes economically unfeasible for the company. Through the *enhancement (or augmentation) effect* generative AI helps employees be more effective, efficient and productive. Hence, although some of the tasks they previously performed would be automated, with the help of generative AI they can complete more tasks, more complex tasks, and more revenue-generating tasks which leads to improved productivity and profitability of the company as a whole. The *transformational effect* changes the nature of work of the employees by creating new tasks for them and/or changing the way they perform tasks.

It should be emphasized that the three effects happen simultaneously. The magnitude of each effect and the balance between them will depend on factors such as the scope of tasks that constitute a job position, their level of automatability, how easy it is to transfer tasks from or to other job positions to compensate for the automated tasks in a particular job position, the technical characteristics and capabilities of the generative AI application, company size and resources, the processes that these tasks are part of, etc. For example, if a job position includes mostly tasks can be easily automated by generative AI with little human intervention to edit AI's output, the implementation of a respective generative AI application in a company would likely result in the elimination of that job position; hence, the substitution effect will prevail. On the other hand, if the tasks in a job position cannot be fully automated, they still require a human to generate content (e.g. by using appropriate prompts) or edit the output (e.g. editing the draft job description or a contract created by AI), the implementation of generative AI is likely to improve the productivity of the employee, make him/her more productive. Hence, the enhancement effect will dominate. Yet, if the generative AI changes the tasks the employee needs to do and the way they have to be performed (e.g. generating images for ads via AI rather than photo shooting), the transformational effect will be stronger than the other two effects. Similarly, in a small company with a few employees, the implementation of generative AI may not lead to significant substitution effect because it might be difficult to transfer the unautomated tasks to other job positions just because there are not enough job positions to transfer them to. Conversely, in a large company, the substitution effect would be much stronger and with tangible outcomes for many employees due to the sheer size of the company—there are many tasks that will be automated and many jobs that may go with them.

In the future, the advancement of the technical characteristics of generative AI will shift the balance in favour of the substitution effect, because many cognitive tasks that are currently unautomatable will become automatable. Employees whose job

positions include such tasks will need to learn, unlearn and relearn how to perform the tasks to remain competitive on the labour market. Through the substitution, enhancement and transformational effects generative AI and human employees will creatively co-destruct jobs by eliminating tasks and jobs, changing the essence and content of tasks and jobs, and creating new tasks and jobs, which will be subsequently eliminated and changed by AI and new ones will be created.

5 Economic Limitations of Generative AI

While generative AI will have a strong positive economic impact on companies that adopt it, there are limitations to its economic contribution to companies, similar to other automation technologies, that go beyond the costs for subscribing and using generative AI applications. The two most important limitations relevant to generative AI are Polanyi's paradox and the Ironies of automation.

Polanyi's paradox was formulated by [20] based on Polanyi's ([21]: 4) observation that "We know more than we can tell.". Every company and organisation has two types of knowledge—codified and tacit knowledge. Codified knowledge is the knowledge in written form in the standard operations manuals of companies, product/service specifications, procedures, process flowcharts, internal rules and regulations, and other documents that explain who does what, how and when. Tacit knowledge is the unwritten knowledge in organization that is shared by its members and reflects their experience how things are actually done in the company in regard to procedures, processes, good practices, power politics, decision-making, etc. The use of automation technologies requires codified knowledge about the processes and decision-making that would be incorporated in the algorithms. Specifically, ([20]: 135) point out that "Engineers cannot program a computer to simulate a process that they (or the scientific community at large) do not explicitly understand". Therefore, tacit knowledge presents a clear barrier to automating processes. The more the tacit knowledge in an organization and less the codified knowledge, the more difficult it is to automate processes. In the context of generative AI, Polanyi's paradox relates to the knowledge how to generate content (text, pictures, code, video), what software applications and prompts to use, how to incorporate Large Language Models into chatbots, how to utilize the generated content into marketing materials, when to use and when not to use generative AI, etc. As evident, much of this knowledge is not so technical (how to use the respective generative AI application) but what to do with its generated content, when and whether to use it. While some of this knowledge can be codified (e.g. in the design of an LLM-based chatbot), other knowledge such as what prompts to use or how to incorporate effectively and efficiently the generated content in marketing communications will have a significant tacit component based on the experience and skills of the human employees that use the generative AI application or its output. Therefore, while AI can automate some of the creative tasks and processes in a company, many other tasks will remain unautomated by it.

Hence, generative AI leads to partial rather than full automation of processes, and human employees are still needed.

Closely related to the Polanyi's paradox are what [22] calls 'the ironies of automation'. In essence, the term denotes a situation when automation increases rather than decreases the work of human employees (i.e. somewhat negative substitution and enhancement effects) due to two main reasons—(a) mistakes in the design of the automated process, and (b) due to the partial automation and the transfer of tasks across job positions, employees may remain with an arbitrary collection of tasks that will actually decrease rather than increase their efficiency and productivity. Therefore, it may turn out that the use of generative AI creates more work for human employees instead of alleviating them. Thus, preference needs to be given to full automation with the use of generative AI and other technologies (as relevant to the specific process) rather than partial automation with generative AI only.

6 Micro- and Macroeconomic Implications

Generative AI has significant micro- and macroeconomic implications beyond the boundaries of the companies that adopt it.

6.1 Microeconomics Implications

- *Market structure*—generative AI allows companies that adopt it to be more competitive. In time, they will increase their market share compared to other companies that do not adopt it. Considering that adopting generative AI does not require prohibitive amount of investment, new companies can easily implement generative AI in their operations, marketing and other business processes. However, considering the economies of scale and synergies that generative AI provides (e.g. using a language model for a chatbot, for creating marketing texts, and for translations), it is likely that it would lead to a long-tail distribution of market shares of companies. Early adopters and large companies that use generative AI in various processes and tasks may have better performance metrics (market share, profitability, customer base, etc.) in the long run compared to the late majority and laggards, and small companies. Therefore, there might be a concentration of market power in a few large companies in an industry. This impact, however, will depend on the role of generative AI in the market competitiveness of companies. For some, e.g. marketing agencies, online travel agencies and other businesses that have high level of automatability of tasks through generative AI, this effect will be much stronger compared to sectors such as agriculture, where generative AI might be of more limited application. In summary, on the one hand, the low barriers to adopt generative AI stimulate its implementation in

many organisations but the economies of scale and synergies may concentrate the market power in a few competitors in the industry.

- *Market prices* for industries with high level of automatability of tasks through generative AI are likely to increase in the short run due to initial expenses to adopt generative AI applications and employee training. In the long run, the experience curve would drive costs down because employees will learn when and how to use the generative AI effectively and efficiently. This tacit knowledge coupled with the economies of scale, in the long run, may lead to a decrease in the market prices in these industries because the implementation of generative AI will decrease their costs (e.g. a digital human in a video would be cheaper than a human actor). Although the actual nominal prices will increase in time due to inflation, the quality and value customers receive will increase much more. Hence, the relative prices per constant quality of the output will decrease.
- *Sources of competitiveness*—the low barriers to adopt generative AI applications mean that access to generative AI will not be a competitive advantage per se but the technical characteristics of the respective generative AI applications used and the processes and tasks they are used for. An application that creates better images or provides more relevant answers without or with minimal hallucination [6] will give an advantage to companies that implement it. Therefore, it is not only the technology that makes companies competitive but how well this technology is utilized. At the same time, a ‘made by humans’ market strategy may be the counterpoint to the AI tsunami—companies can offer products and services manufactured/delivered only by humans to distinguish themselves as ‘high touch’ companies that offer authentic services [23] on the predominantly ‘high tech’ market.
- *New markets*—generative AI creates new products, services, and essentially new markets. Companies and customers can buy, sell or rent digital humans and avatars, clothes for them, etc. For example, companies that offer digital avatars for video presentations, provide basic templates and a handful of avatars with basic functionalities free of charge while access to more advanced features, avatars and templates is charged.

6.2 Macroeconomics Implications

- *Job displacement*—as already elaborated, although generative AI creates new jobs and transforms existing jobs, its substitution effect in the long run will prevail due to the technological advances. Hence, in the short run more jobs may be created than displaced but the long-term prospects may not be in favour of labour and the substitution effect may dominate.
- *Digital divide* between countries, industries, companies, and employees who actively use generative AI in their daily operations and reap the benefits of the technology, and those who do not do it due to the lack of knowledge, skills, financial resources, or interest/motivation. Considering that generative AI will

make companies more competitive, such digital divide will result in economic disparities as well.

- *Economic disparities*—the economic power, and respectively wealth, will be concentrated in companies that produce widely-adopted generative AI applications and their large clients. The same is valid for managers and employees—those who have the knowledge and skills to use effectively and efficiently generative AI will be of high demand on the labour market and better paid than those who lack these digital skills.
- *Economic growth*—by fostering innovations, competitiveness, new products, services and markets, generative AI ultimately stimulates gross value added and economic growth. How this growth is distributed among economic players (industries, companies, employees) is a different question.

7 Concluding Remarks

Generative AI is a technological tool, sophisticated but a tool. As elaborated in the chapter, managers and owners need to perform a comprehensive cost-benefit analysis to support their decision to adopt generative AI or not. While there are significant benefits of implementing generative AI, costs are not to be underestimated. Additionally, managers need to remember that the implementation of generative AI has substitution, enhancement and transformational effects on jobs simultaneously. The costs, benefits and the balance between the three effects are specific to each company and depend on its industry and size, the organisation of its internal business processes, the automatability of tasks that constitute them and other factors. While in the short term the enhancement (augmentation) and transformational effects dominate, in the long term it would be the substitution effect that would be strongest. Of course, this does not mean that people will ‘go the way of the horses’ (to answer [24]’s question) but what they do and how they do it will inevitably change [17]. In fact, it will not be generative AI that will replace employees but employees who use AI will replace employees who do not use AI. We can even go further and generalize that companies that use AI will replace companies that do not. As with other technologies, some resistance, skepticism and even fear among managers, employees and customers towards generative AI are inevitable. In time, its wider implementation and the greater exposure of people to it may increase their awareness about its capabilities, its real rather than assumed benefits and costs to companies, and its actual substitution, enhancement and transformational effects which may change people’s perceptions.

The economic aspects of technology in general and of AI in particular are often overlooked. Hype/fashion, herd behavior and inertia may shroud a sober economic appraisal of generative AI adoption. The facts that generative AI catches the public’s imagination, that other companies adopt it or that the company has established procedures that do not involve AI are not genuine reasons to adopt generative AI or not;

they are excuses for not performing an in-depth economic analysis. The lack of analysis may disappoint managers and owners because the implementation of AI may not match their overly optimistic expectations (see also [25]).

Future research needs to delve into the actual costs and benefits that generative AI brings to companies through case study research of companies that successfully adopted it or that failed to do it in order to identify the critical success factors of generative AI adoption. Empirical research may shed light on the actual substitution, enhancement and transformational effects and the factors that drive them. Research may also measure empirically the micro- and macroeconomic effects of generative AI.

References

1. Pavlik, J. V. (2023). Collaborating with ChatGPT: Considering the implications of generative artificial intelligence for journalism and media education. *Journalism and Mass Communication Educator*, 78(1), 84–93. <https://doi.org/10.1177/10776958221149577>
2. Carvalho, I., & Ivanov, S. (2023). ChatGPT for tourism: Applications, benefits, and risks. *Tourism review*. (in press). <https://doi.org/10.1108/TR-02-2023-0088>
3. Dwivedi, Y. K., Kshetri, N., Hughes, L., Slade, E. L., Jeyaraj, A., Kar, A. K., Baabdullah, A. M., Koohang, A., Raghavan, V., Ahuja, M., Albanna, H., & Wright, R. (2023). So what if ChatGPT wrote it? Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy. *International Journal of Information Management*, 71, 102642. <https://doi.org/10.1016/j.ijinfomgt.2023.102642>
4. Fusté-Forné, F., & Orea-Giner, A. (2023). Gastronomy in tourism management and marketing: an interview with ChatGPT. *ROBONOMICS: The Journal of the Automated Economy*, 4, 42. Retrieved from <https://journal.robonomics.science/index.php/rj/article/view/42>
5. Iskender, A. (2023). Holy or Unholy? Interview with open AI's ChatGPT. *European Journal of Tourism Research*, 34, 3414. <https://doi.org/10.54055/ejtr.v34i.3169>
6. Ivanov, S., & Soliman, M. (2023). Game of algorithms: ChatGPT implications for the future of tourism education and research. *Journal of Tourism Futures*, 9(2), 214–221. <https://doi.org/10.1108/JTF-02-2023-0038>
7. Lim, W. M., Gunasekara, A., Pallant, J. L., Pallant, J. I., & Pechenkina, E. (2023). Generative AI and the future of education: Ragnarök or reformation? A paradoxical perspective from management educators. *The International Journal of Management Education*, 21(2), 100790. <https://doi.org/10.1016/j.ijme.2023.100790>
8. Sun, J., Liao, Q. V., Muller, M., Agarwal, M., Houde, S., Talamadupula, K., & Weisz, J. D. (2022, March). Investigating explainability of generative AI for code through scenario-based design. In *27th international conference on intelligent user interfaces* (pp. 212–228). <https://doi.org/10.1145/3490099.3511119>
9. DiMaggio, P. J., & Powell, W. W. (1983). The iron cage revisited: Institutional isomorphism and collective rationality in organizational fields. *American Sociological Review*, 48(2), 147–160.
10. Eisfeldt, A. L., Schubert, G., & Zhang, M. B. (2023). *Generative AI and firm values* (No. w31222). National Bureau of Economic Research. Retrieved from https://www.nber.org/system/files/working_papers/w31222/w31222.pdf
11. Ivanov, S., & Webster, C. (2018). Adoption of robots, artificial intelligence and service automation by travel, tourism and hospitality companies—A cost–benefit analysis. In V. In Marinov, M. Vodenska, M. Assenova, & E. Dogramadjieva (Eds.), *Traditions and Innovations in Contemporary Tourism* (pp. 190–203). Cambridge Scholars Publishing.

12. Ivanov, S., & Webster, C. (2019). Economic fundamentals of the use of robots, artificial intelligence and service automation in travel, tourism and hospitality. In S. Ivanov, & C. Webster, (Eds.) *Robots, artificial intelligence and service automation in travel, tourism and hospitality* (pp. 39–55). Emerald Publishing. <https://doi.org/10.1108/978-1-78756-687-320191002>
13. Müller, V. C. (2021). Ethics of artificial intelligence and robotics. In E. N. Zalta, (Ed.) *The Stanford encyclopedia of philosophy* (Summer 2021 Edition). Retrieved from <https://plato.stanford.edu/archives/sum2021/entries/ethics-ai/>
14. Baker-Brunnbauer, J. (2021). TAII framework for trustworthy AI systems. *ROBONOMICS: The Journal of the Automated Economy*, 2, 17. Retrieved from <https://journal.robonomics.science/index.php/rj/article/view/17>
15. Coeckelbergh, M. (2020). *AI ethics*. MIT Press.
16. Bornet, P., Barkin, I., & Wirtz, J. (2021). *Intelligent automation: Welcome to the world of hyperautomation*. World Scientific.
17. Webster, C., & Ivanov, S. (2020). Robotics, artificial intelligence, and the evolving nature of work. In B. George, & J. Paul (Eds.). *Digital transformation in business and society theory and cases* (pp. 127–143). Palgrave-MacMillan.
18. Deschacht, N. (2021). The digital revolution and the labour economics of automation: A review. *ROBONOMICS: The Journal of the Automated Economy*, 1, 8. Retrieved from <https://journal.robonomics.science/index.php/rj/article/view/8>
19. Ivanov, S. (2020). The impact of automation on tourism and hospitality jobs. *Information Technology and Tourism*, 22(2), 205–215. <https://doi.org/10.1007/s40558-020-00175-1>
20. Autor, D. H. (2015). Polanyi's Paradox and the shape of employment growth. In: Re-evaluating labor market dynamics (pp. 129–177). Federal Reserve Bank of Kansas City.
21. Polanyi, M. (1966). *The tacit dimension*. Doubleday.
22. Bainbridge, L. (1983). Ironies of automation. *Automatica*, 19(6), 775–779.
23. Seyitoğlu, F. (2021). Automation versus authenticity in services. *ROBONOMICS: The Journal of the Automated Economy*, 2, 20. Retrieved from <https://journal.robonomics.science/index.php/rj/article/view/20>
24. Brynjolfsson, E., & McAfee, A. (2015). Will humans go the way of horses? Labor in the second machine age. *Foreign Affairs*, 94(4), 8–14.
25. Ivanov, S. (2022). The economics of technology in travel, tourism, and hospitality. *Journal of Global Hospitality and Tourism*, 1(2), 175–177. <https://doi.org/10.5038/2771-5957.1.2.1013>

Plant Data Generation with Generative AI: An Application to Plant Phenotyping



Swati Bhugra, Siddharth Srivastava, Vinay Kaushik, Prerana Mukherjee, and Brejesh Lall

Abstract Plant phenotyping is the study of plants' physiological, morphological and biochemical traits resulting from their interaction with the environment. These traits (e.g., leaf area, leaf count, tillering, wilting etc.) are crucial in current plant research, focused on improving plant quality i.e., disease resistance, drought resistance and productivity. With the advancement in sensor technologies, image based analysis via various computer vision methods (e.g., image classification, segmentation, object detection etc.) have emerged in plant phenotyping. Specifically, state-of-the-art deep learning models have been employed for high-throughput study of plant traits. However, the application of deep learning models is currently limited due to the high variability in plant traits among various plant species and unstructured plant imaging. Additionally, complex plant traits pose high data collection and annotation costs. In this context, generative artificial intelligence (AI) based on the evolution of generative adversarial networks (GANs) for data synthesis can relieve the current bottleneck of data scarcity and plant species gap. This chapter reviews the application of state-of-the-art GANs for plant image datasets such as leaf, weed, disease etc. It also discusses the current Generative AI challenges and future directions for agriculture data synthesis.

Keywords Plant phenotyping · Plant phenomics data · Generative artificial intelligence (AI) · Generative Adversarial Networks (GANs) variants · Disease segmentation · Leaf segmentation

1 Introduction: Plant Phenomics

Plant phenotyping is the comprehensive study of plant traits (e.g., leaf area, biomass, wilting, tillering etc.) based on their interaction with the environment [1]. These traits are termed as plant phenotypes and the discipline is termed as plant phenomics. By systematically analysing plant the phenotypes, biologists can uncover the underly-

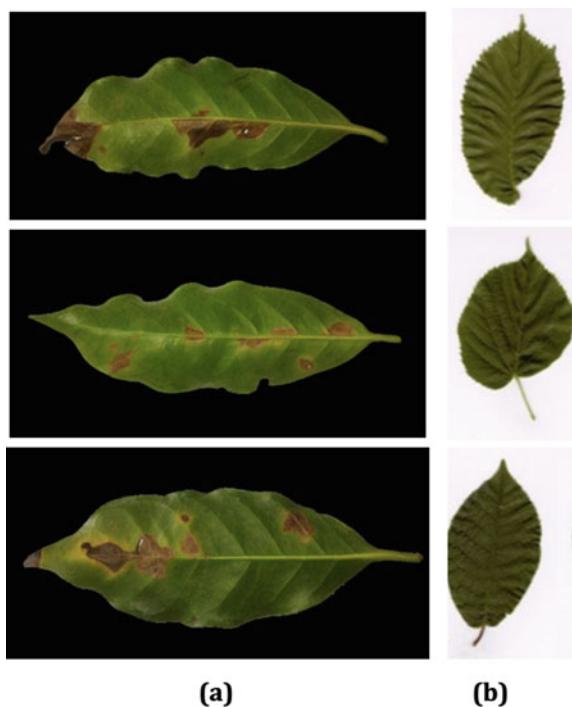
S. Bhugra (✉) · S. Srivastava · V. Kaushik · P. Mukherjee · B. Lall
Indian Institute of Technology, New Delhi, India
e-mail: brejesh@ee.iitd.ac.in

ing mechanisms that govern the interactions of plant traits with its environment. Thus, plant phenotyping plays a crucial role in advancing plant biology research and facilitating plant quality control i.e., disease resistance, drought resistance etc. [2]. Traditionally, biologists relied on manual screening of plant traits resulting in highly subjective and low-throughput phenotyping.

The advancement in sensor technologies permits high-throughput monitoring of plants at different growth stages using imaging technologies, such as visible imaging, hyperspectral imaging, the mal imaging etc. [3]. Based on plant image data, image based analysis via vision computer vision methods (e.g. image classification, segmentation, object detection etc.) have emerged in plant phenotyping [2].

It is to be noted that high biological variability for example variance in disease symptoms (Fig. 1(a)) pose significant challenges in designing hand-crafted features for reliable plant phenotyping (in this example, disease classification) [4]. In contrast, deep learning models exploit the raw image data without relying on expert biologists for formulating hand-crafted features. Thus, state-of-the-art deep learning models have been proposed for the extraction of various plant phenotypes such as, leaf area via leaf segmentation (object segmentation models), leaf counting via leaf detection (object detection models) and stress responses via disease detection (image classification models) [7]. However, high variance in plants' phenotypes among various plant species and unstructured plant imaging (Fig. 1) limits the generalizability of

Fig. 1 Sample plant images highlighting the biological variability **a** disease symptoms [5] and **b** leaf shape [6]



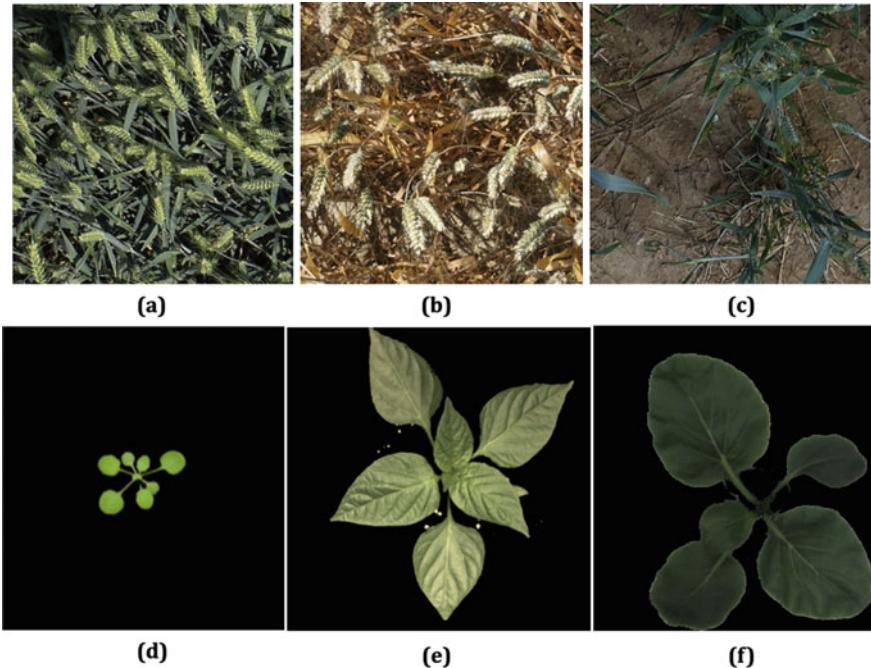


Fig. 2 Sample plant images highlighting the complex plant phenotypes **a, b, c** wheat spike in field conditions [9] and **d, e, f** leaf occlusion [15–17]

deep learning models. Secondly, complexity of plant phenotypes (Fig. 2) demands costly data acquisition protocols and annotations. Lastly, with limited research collaborations among different fields (computer vision, agriculture, robotics etc.) in plant phenotyping have resulted in few public datasets (Table 1). It is to be noted that the majority of the annotated dataset publicly available have been introduced as computer vision challenges (CVPNP). The cost of annotating large image dataset is evident from contrasting the pixel-level annotation provided for small-scale coffee dataset [5] and only image-level annotation provided for large-scale PlantVillage dataset [8] (Table 1). Similarly, complex phenotypes and unstructured imaging limits the instance level annotations of multiple spikes present in wheat spike dataset [9].

It is to be noted that various studies in deep learning have empirically shown that large datasets lead to optimal model training whereas small datasets may result in overfitting [10, 11]. Thus, limited datasets are the current bottleneck in image based plant phenotyping. Although public dataset are available in the computer vision community such as ImageNet [12], MSCOCO [13], PASCAL VOC [14] that consists of approximately million annotated images, the features learnt on these datasets lack transferability to plant image data. This is because the plants' phenotypes exhibit deformability unlike objects such as table, chair etc. in the computer vision datasets.

In this context, algorithms focused on data synthesis provide a prime solution. The evolution of generative adversarial networks (GANs) [18] has led to the emergence

Table 1 Publicly available plant image datasets with various annotation levels, here N/A refers to not applicable

Dataset	Data	Image	Annotations			
			Classification	Detection	Segmentation	Instance
Sorghum Biomass 1	~277,327	Field	N/A	No	Yes	No
Wheat Spike 2	~1722	Field	N/A	Yes	Yes	No
Root 3	~207	Control	N/A	No	Yes	Yes
Arabidopsis Canopy 4	~284	Control	N/A	No	Yes	Yes
MSU-Bean Canopy 5	~350	Control	N/A	No	No	Yes
Salad Canopy 6	~75	Control	N/A	No	No	No
Komutsana Canopy 7	~300	Control	N/A	No	No	Yes
Coffee Leaf Disease	~1747	Control	Yes	No	Yes	N/A
PlantVillage Disease 8	~54306	Control	Yes	No	No	N/A
Maize NLB Disease 9	~1756	Field	Yes	No	No	No

of generative artificial intelligence (AI) with an objective to generate new instances based on patterns and structures learned from the original dataset. By leveraging generative AI, researchers can augment the limited datasets in plant phenomics with artificially generated data. Thus, providing a substantial boost to the experimental capacity and expanding the range of phenotypes that can be explored [19].

The rest of the chapter is organised as follows: in Sect. 2, we review the traditional data augmentation methods employed in plant phenotyping. Based on the aforementioned advantages of Generative AI in plant phenomics, in Sect. 3 we explain the evolution of Generative AI with respect to different GAN based architectures. In Sect. 4, we review the GANs employed in plant phenomics. This is followed by a section on evaluation metrics that are commonly employed to quantitatively contrast GAN based architectures (Sect. 5). Lastly, we highlight the current limitations in Generative AI with its future applications in plant phenomics (Sect. 6). By shedding light on the limitations of generative AI, we hope to encourage further research and development in this emerging field and contribute to the advancement of plant research studies.

2 Traditional Data Augmentation Methods in Plant Phenomics

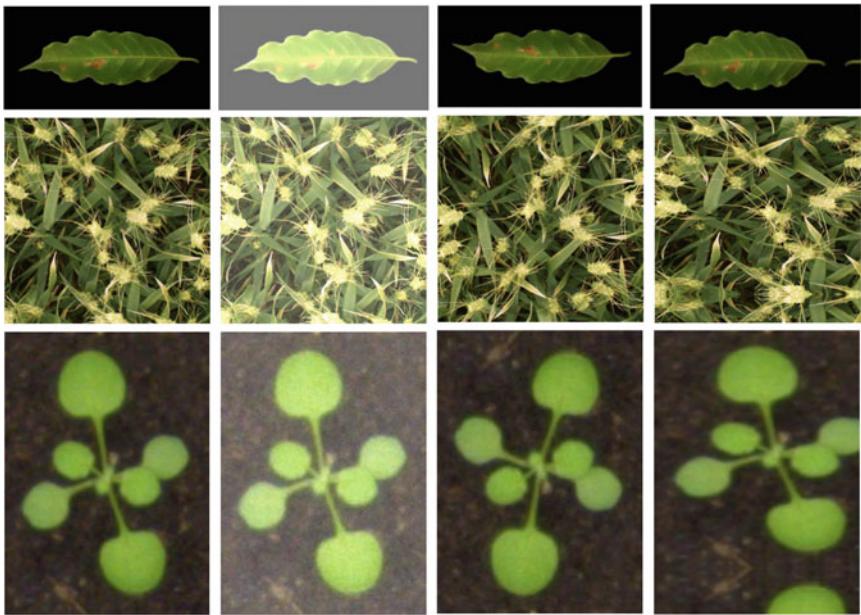
The establishment of phenotyping platforms facilitated image based screening of plants. Based on this data, various computer vision methods have been employed for non-invasive plant phenotyping [20]. Specifically, current image based plant phenotyping methods are primarily formulated as supervised deep learning tasks such as leaf segmentation, leaf counting etc. However, the applicability of these models is still limited due to the scarce dataset (Table 1). To relieve this bottleneck, early plant phenotyping studies employed traditional augmentation methods such as rotation, scaling, graphical modeling etc. to augment plant image data. For example, authors [21–23] employed these augmentation methods at the training stage of deep learning models for supervised leaf segmentation task. Similarly, the accuracy of disease segmentation task was improved in [24, 25] via training with augmented data. In the following subsections, we discuss traditional data augmentation methods in plant phenotyping categorised as (a) Basic augmentation methods, (b) Cut and paste [26] methods and (c) Graphical modeling [27].

2.1 Basic Augmentation Methods

These methods include basic geometrical or colour transformation. Geometrical transformation methods include rotations, translations, scaling, flipping, and cropping [28, 29]. These transformation methods modify the spatial orientation, position, and scale of plant images. In contrast, color transformation [29–31] on plant images refers to modification of its color channels. These methods provide variance with respect to lighting conditions or imaging set-ups and can improve the robustness of deep learning models to variations in image quality. In addition, noise injection such as Gaussian noise, random pixel dropout etc. [32] have also been used to simulate imperfections encountered in unstructured field images. It has been shown to improve the robustness of the deep learning models to noise [33, 34]. Figure 3 shows sample images generated with the aforementioned methods.

2.2 Cut and Paste Augmentation Method

This method is primarily based on the utilisation (cutting) of object instances such as leaf, spike etc. and placing (pasting) them at known positions in a random background for data synthesis. This permits rapid and automatic generation of training data for instance detection and segmentation tasks. For example, authors in [35] employed this technique to synthesise dataset that mimics the characteristics of Computer Vision Problems in Plant Phenotyping (CVPPP) dataset [15] such as top-view images



(a) Input Image (b) Random Brightness (c) Random Flip (d) Random Distortion

Fig. 3 Sample images generated with basic augmentation methods, First row: Coffee leaf disease dataset [5], Second row: wheat spike dataset [9] and Third row: plant canopy dataset [15]

with similar backgrounds (plant pots) and leaves emerging from the centre. The authors in this study trained a deep learning model with generated synthetic data and 10% of CVPDP dataset, surpassing the benchmark performance on leaf instance segmentation task. Similar studies [36, 37], showcased the effectiveness of the cut and paste augmentation method in plant phenotyping. Figure 4a shows spike instances copy pasted on a random background. The red boxes shows the background region in Fig. 4b pasted with spike instances in Fig. 4c.

2.3 Graphical Modeling Based Augmentation Method

The Lindenmayer systems (L-systems) [39], theoretically represented plant growth via geometric interpretations and have been widely used to synthesis plant models. For example, L-peach model was proposed by Allen et al. [40] to virtually simulate tree growth stages via various structural and physiological transformations. Similarly, Leitner et al. [41] employed heuristics to model the growth of plant roots using L-system. To calibrate developmental plant models for maize and canola, Cieslak et al. [42] introduced an L-system based interactive approach. With the advancement in research for plant growth systems [43], authors in [44] combined the plant archi-

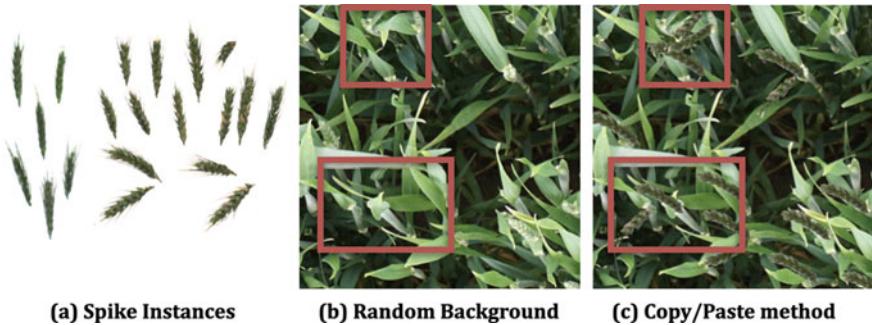


Fig. 4 **a** Spike instance dataset [38], **b** Random background and **c** Sample image using copy/paste method

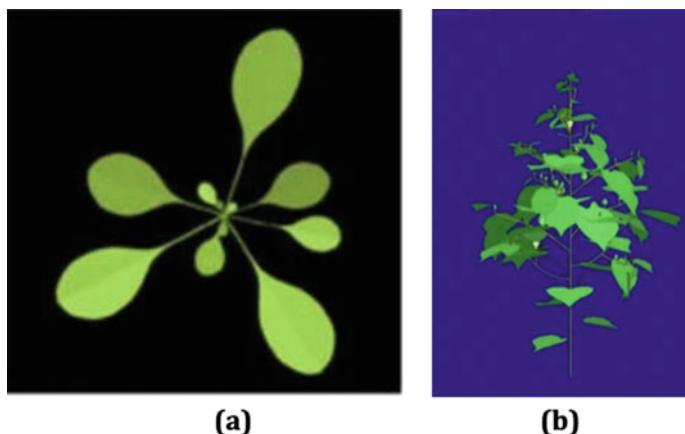


Fig. 5 Virtual plants simulated via graphical modeling in **a** [27] and **b** [44]

tectural parameters with physiological parameters to virtually simulate the cotton growth. Similarly, Espana et al. [45] implemented a two-dimensional rectangular plane to map different parts of the 3D maize leaf blade permitting its virtual 2D and 3D morphological simulation.

Recently, L-system based plant simulator software fitted with probabilistic curves derived from different plant phenotypic traits have been developed to model Arabidopsis in [27]. The proposed method augments the training data by generating synthetic leaves in a leaf-by-leaf manner. Similarly, Ward et al. [46] followed a leaf-by-leaf approach for leaf and texture synthesis (extracted from CVPPP dataset [15]) for overall plant canopy generation. By augmenting the training data with synthetic images generated with the proposed pipeline, the authors achieved a noteworthy improvement on tobacco dataset [15] that exhibits dense leaf occlusions compared to Arabidopsis dataset. Graphical modeling method provides flexibility in modeling different plant phenotypes (such as leaf) but is limited in terms of its morphological,

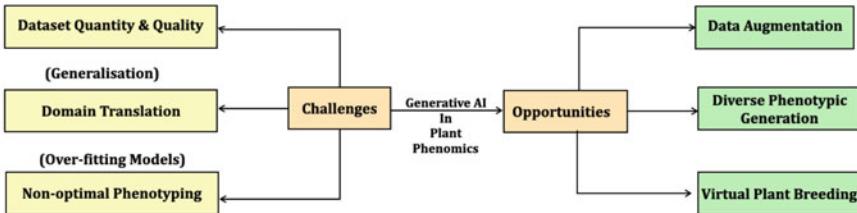


Fig. 6 Challenges in Plant Phenomics and Opportunities for Generative AI

color, and texture representation of real plant data. Figure 5 shows sample augmented images via graphical modeling methods [27, 44].

To summarise, the limitations of traditional data augmentation methods are:

- The datasets from these augmented methods inherits similar bias present in the original datasets. For example due to the symmetrical nature of plant canopy images (Fig. 3c), random flip may be a redundant transformation with no significant change to the arrangement of plant leaves.
- Selecting suitable augmentation methods for a dataset can be a challenging task. For instance, color transformations may interfere with disease color characteristics (brown, yellow spots) thus changing the semantics for reliable disease classification (Fig. 3b).
- The cut/paste and graphical modeling methods are dependent on pre-defined parameters such as numbers of leaves, rotation angles, and leaf sizes that greatly varies with different plant species and requires expert knowledge.
- Quality control of the augmented dataset is a costly endeavour [47].

In contrast to the aforementioned transformations that rely only on a single image, data augmentation based on Generative Adversarial Networks (GANs) have been shown to provide large data variance. Specifically, it enables the generation of large-scale diverse datasets, thereby facilitating more comprehensive and representative analysis of plant phenotypic traits. Secondly, it address the challenge of labelled data in plant phenotyping. The ability to generate synthetic labelled samples significantly reduces the dependence on manual annotations, that can be a laborious task and prone to human error. These generated labelled data with varying degrees of complexity, enables optimal training of deep learning models (Fig. 6). Thus, we aim to explore Generative AI algorithms employed in plant phenotyping with special focus on GANs architectures, discussed in the next section.

3 Generative Artificial Intelligence (Generative AI)

Generative AI is an emerging field in artificial intelligence with the focus on development of algorithms and models that learns from large training data to synthesize

novel content. Specifically, these models can be trained to understand patterns, structures, and styles inherent in the data enabling novel data synthesis aligned with the learned features.

The introduction of Generative Adversarial Networks (GANs) [18] represents a paradigm shift in this field and since then has advanced with the evolution of different GAN architecture. Thus, in the next subsection, we firstly elucidate the basic principles of GANs, followed by short descriptions of its variants.

3.1 Generative Adversarial Networks (GANs)

The basic architecture comprises of two neural networks, a generator and a discriminator (Fig. 7). The generator starts by accepting random noise as input and uses a neural network to transform the input noise into a synthetic sample similar to the original data. The generated sample is then passed to the discriminator, another neural network to evaluate the authenticity of the generated data. Specifically, the training of GANs involves an iterative optimization process where the generator and discriminator networks are updated alternately [48]. During each iteration, the generator network generates synthetic samples, which are then evaluated by the discriminator. The discriminator provides feedback to the generator by assigning a probability score indicating the likelihood of the generated samples being real. This feedback is used to compute the gradients for updating the generator's parameters, allowing it to generate more samples that are indistinguishable from the original data. Simultaneously, the discriminator's parameters are updated based on its ability to correctly classify real and synthetic samples. The generator and discriminator continue to be trained in an adversarial manner until the generator network learns to synthesize data to fool the discriminator. Typically, a generator learns latent vectors that represent different features of the training data to generate coherent synthetic data.

One of the key benefits of GANs is their ability to produce high-quality realistic images. For example, authors in [49] trained GAN on celebrity faces dataset, and

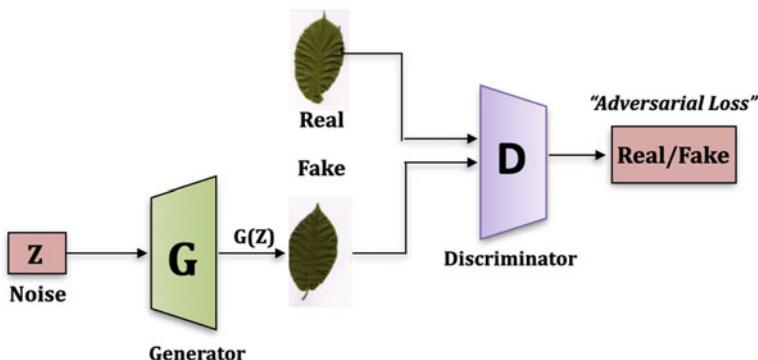


Fig. 7 The framework of Generative Adversarial Networks (GANs)

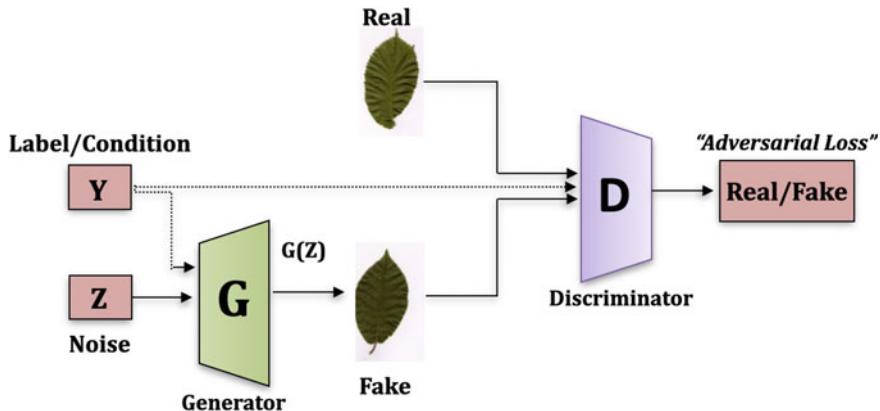


Fig. 8 The framework of Conditional Generative Adversarial Networks (cGANs)

efficiently generated highly realistic images of faces not present in the training data. This demonstrates the ability of GANs to capture the underlying distribution of the data, and to generate novel synthetic samples that are similar to the real data. Fig. 8

3.2 Conditional Generative Adversarial Networks (cGANs)

Conditional Generative Adversarial Networks (cGANs) [50] combine the power of Generative Adversarial Networks (GANs) based on an input label to conditionally generate realistic data samples. cGANs comprises of two primary components: a generator network and a discriminator network. The generator network takes as input a random noise vector typically drawn from a probability distribution such as Gaussian or uniform distribution, along with the conditioning information. The conditioning information can be any form of auxiliary input, such as class labels, attributes, or other relevant metadata [51]. The generator then processes this input and attempts to produce synthetic samples that resemble the real data distribution conditioned on the given input. On the other hand, the discriminator network aims to distinguish between real and synthetic samples. Through an adversarial training process, the generator network learns to generate samples that fool the discriminator, while the discriminator learns to become more proficient in discriminating between real and synthetic samples.

The objective function of cGANs combines both the adversarial loss and the conditioning loss. The adversarial loss measures the ability of the generator to deceive the discriminator, while the conditioning loss ensures that the generated samples conform to the given conditioning information. By jointly optimizing these loss functions, cGANs can produce high-quality samples that not only capture the underlying data distribution but also align with the desired conditional information.

One of the key advantages of cGANs is their ability to generate data samples that are conditioned on specific attributes or labels. This makes them particularly useful for tasks such as image-to-image translation [52], where the generator can be conditioned on an input image to generate a corresponding output image with desired attributes or characteristics.

3.3 Deep Convolutional Generative Adversarial Networks (DCGANs)

Deep Convolutional Generative Adversarial Networks (DCGANs) [49] leverage the power of Convolutional Neural Networks (CNNs) [53] to generate visually coherent and detailed images by learning from a large dataset of real images. The architecture of DCGANs consists of two main components: a generator network and a discriminator network.

The generator network typically starts with a low-dimensional random noise vector and gradually upsamples it through a series of deconvolutional layers. These layers perform a process known as transpose convolution or fractionally strided convolution, which expands the spatial dimensions of the input noise to match the desired output image size [54]. The deconvolutional layers are often accompanied by batch normalization and non-linear activation functions, such as ReLU or LeakyReLU [55], to enhance the stability and expressiveness of the generator network (Fig. 9). The discriminator network, akin to a traditional CNN classifier [56], consists of a series of convolutional layers followed by fully connected layers. The convolutional layers extract hierarchical features from the input image and gradually reduce its spatial dimensions. The discriminator's objective is to correctly classify whether an image is real or generated. It learns to distinguish the subtle differences between real and synthetic images by leveraging the convolutional layers' ability to capture local spatial patterns and structures.

The training process is guided by an objective function that combines both the adversarial loss and the reconstruction loss, which measures the similarity between the generated images and the real images. To stabilize the training process and improve the quality of the generated images, several architectural guidelines are commonly followed in DCGANs. These guidelines typically include (a) using strided convolutions in the discriminator to progressively downsample the image spatially, (b) using batch normalization to normalize the intermediate activations, (c) avoiding fully connected layers inside the convolutional networks, and (d) employing specific weight initialization strategies [55].

3.4 Cycle-Consistent Generative Adversarial Networks (CycleGANs)

Cycle-Consistent Generative Adversarial Networks (CycleGANs) [57] is designed for unsupervised image-to-image translation. Unlike traditional methods that require paired training data, CycleGANs can learn to map images from one domain to another without explicit correspondences.

The cycle consistency principle exploits the idea that if an image from domain X is translated to domain Y and then translated back to domain X , it should ideally be close to the original image. Similarly, an image from domain Y should return to its original counterpart after being translated to domain X and then back to domain Y . This cycle consistency is enforced by introducing cycle consistency loss, which measures the discrepancy between the original image and the reconstructed image obtained via aforementioned translations.

To implement cycle consistency principle, CycleGANs simultaneously train two pairs of generators and discriminator (Fig. 10). And the training of CycleGANs

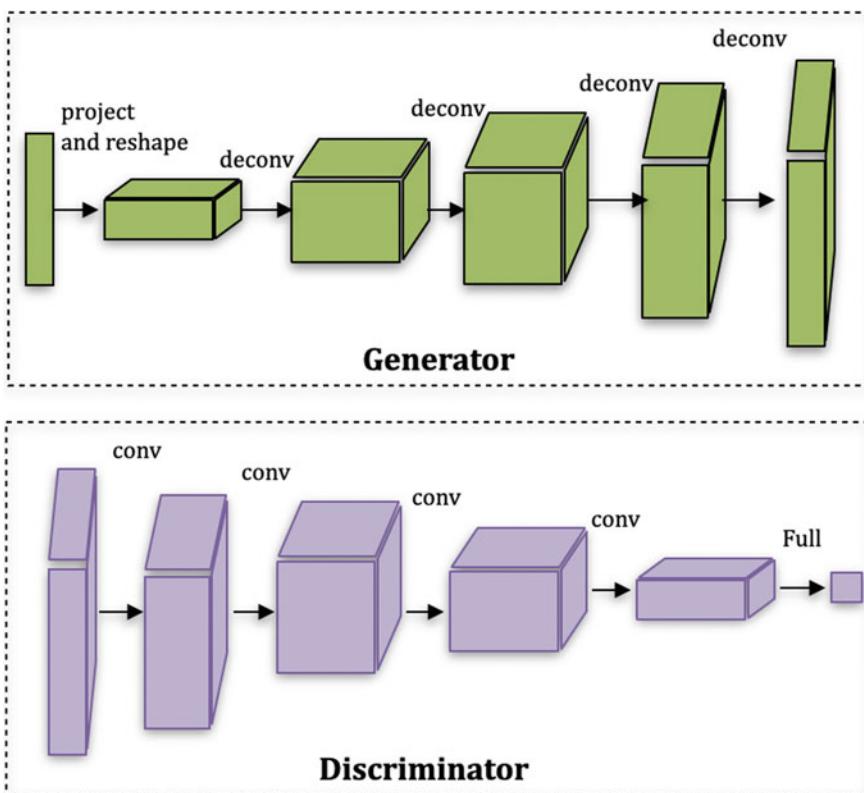


Fig. 9 The DCGANs pipeline, here conv: convolution and deconv: deconvolution

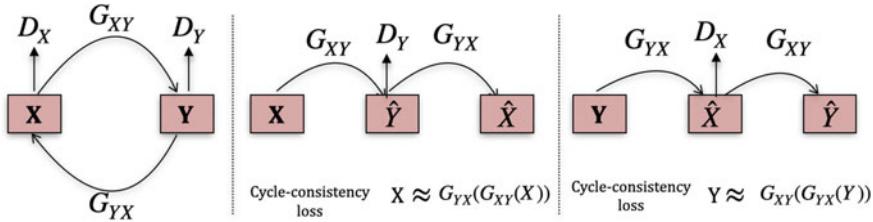


Fig. 10 The training work-flow of CycleGANs

[52] is driven by two main principles: adversarial learning and cycle consistency. Adversarial learning involves a min-max game between the generators and discriminators, where the generators aim to generate realistic images that deceive the discriminators, while the discriminators strive to correctly classify between real and generated images. This adversarial loss encourages the generators to produce high-quality translations that are indistinguishable from real images in the target domain. To achieve cycle consistency, CycleGANs leverage a combination of forward and backward translation paths. The forward generator G_{XY} maps images from domain X to domain Y , while the backward generator G_{YX} maps images from domain Y to domain X . Specifically, the cycle consistency loss for the generator G_{XY} is computed by evaluating the difference between X and $G_{YX}(G_{XY}(X))$, while the cycle consistency loss for the generator G_{YX} is computed based on the difference between Y and $G_{XY}(G_{YX}(Y))$. By incorporating cycle consistency loss, the generators are incentivized to produce translations that preserve important information and characteristics of the original images.

CycleGANs offer a flexible and unsupervised approach to learn mappings between different visual domains without the need for paired training data, making them particularly valuable in scenarios where obtaining paired data is challenging or expensive [58].

3.5 Progressively Growing Generative Adversarial Networks (ProGANs)

Progressively Growing Generative Adversarial Networks (PGGANs) [59] leverage a progressive growing strategy to train high-resolution image generators in a stable and controlled manner. The progressive growing strategy ensures that the network learns coarse features first and then progressively refines the details as the training advances.

The architecture of PGGANs (Fig. 11) consists of a series of generator and discriminator networks, where each network corresponds to a specific resolution level and during training, PGGANs follow a two-step process for each resolution level [60]. In the first step, a lower-resolution generator and discriminator are trained to

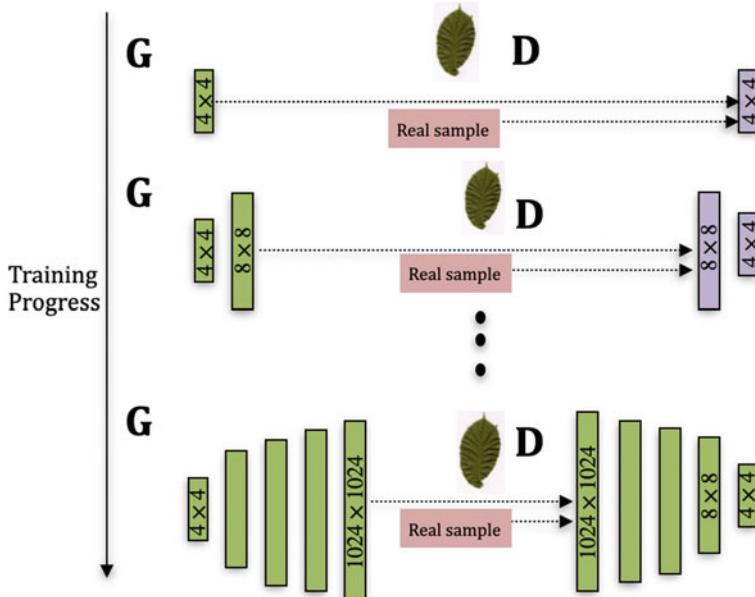


Fig. 11 The training work-flow of Progressively Growing Generative Adversarial Networks (Pro-GANs), here G:Generator and D:Discriminator

capture the coarse features of the target distribution. The generator takes a latent noise vector as input and produces a low-resolution image, while the discriminator aims to distinguish between real and generated images at that resolution level. In the second step, the resolution is increased, and additional layers are added to both the generator and discriminator networks. The weights of the previously trained networks are used as a starting point for the newly added layers. The generator and discriminator are then further trained using higher-resolution images, allowing the network to capture more fine-grained details. The step-by-step training process ensures that the network can gradually learn complex patterns and avoid the instability that can arise when training a large network from scratch.

To ensure stable training and avoid the “mode collapse” problem [61], PGGANs employ a variety of architectural and training techniques. One crucial technique is the use of minibatch standard deviation, which encourages the generator to produce diverse samples by incorporating statistical information from multiple samples in the minibatch. Another technique is the equalized learning rate, which normalizes the weight magnitudes and helps in balancing the training process [62] (Fig. 11).

3.6 Style Generative Adversarial Networks (StyleGANs)

Style Generative Adversarial Networks (StyleGANs) [63] focus on generating images with highly controllable and customizable visual styles. Unlike traditional Generative Adversarial Networks (GANs) that primarily learn to generate realistic images, StyleGANs emphasizes on the generation of images that exhibit specific artistic or stylized features. The key innovation in StyleGANs lies in their ability to separate the control of high-level features, referred to as “styles,” from the low-level details of the generated images. This separation enables the generation of diverse and highly customizable images by manipulating the style parameters.

The architecture of StyleGANs consists of two primary components: a generator network and a discriminator network (Fig. 12). The generator network uses a series of convolutional layers with adaptive instance normalization to transform the input noise and style vectors into the output image. The input noise vector provides the randomness and variability in the generated images, while the style vectors control the high-level features such as colors, textures, and other stylistic attributes. The discriminator network receives both real images and generated images from the generator network as input and learns to accurately classify them.

StyleGANs adopts a unique training strategy called progressive growing [59] (see previous subsection), which involves gradually increasing the resolution of the generated images during training. This approach allows for controlling the high-frequency details while maintaining the stability of the training process [64]. Additionally, a feature matching loss is employed to encourage the generator to match the intermediate feature representations of the discriminator, improving the overall image quality.

3.7 Information Maximizing Generative Adversarial Networks (InfoGANs)

Information Maximizing Generative Adversarial Networks (InfoGANs) [65] extend the traditional Generative Adversarial Networks (GANs) framework by explicitly maximizing the mutual information between selected latent variables and the generated data. While traditional GANs only involve random noise as input to the generator, InfoGANs introduce additional latent variables, referred to as “latent codes,” that can be systematically controlled to manipulate specific attributes or features of the generated data. Specifically, InfoGANs aim to learn disentangled representations of data by encouraging the generator to capture meaningful and interpretable features.

The architecture of InfoGANs consists of a generator network, a discriminator network, and an auxiliary network (Fig. 13). The generator network takes as input a noise vector, as well as the latent codes that represent specific features. The generator transforms these inputs into synthesized data samples that resemble the real data distribution. The discriminator network has a dual role in InfoGANs. It not only dis-

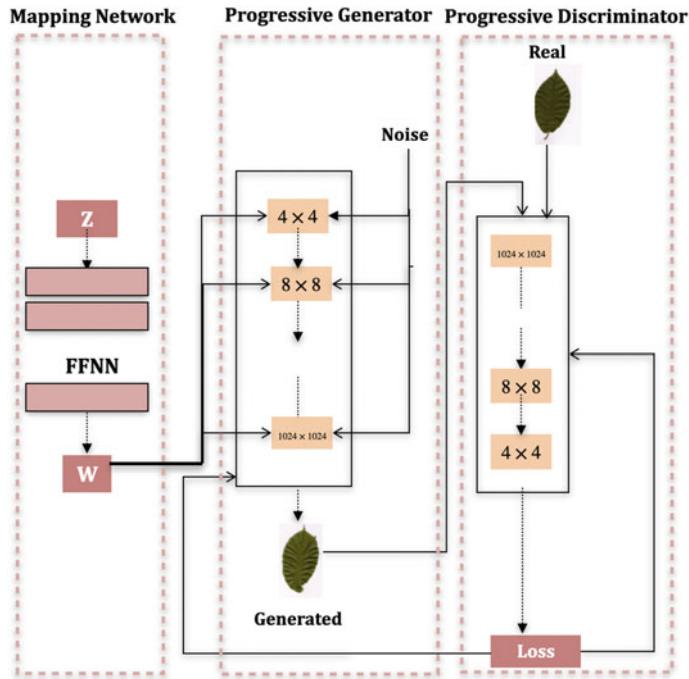


Fig. 12 The framework of Style Generative Adversarial Networks (StyleGANs), here z :latent vector and w :latent space allows controls of “styles” [10]

criminations between real and generated data but also estimates the values of the latent codes associated with the generated samples. The auxiliary network is responsible for estimating the latent codes based on the features extracted from the intermediate layers of the discriminator.

The training process of InfoGANs involves an adversarial game between the generator and discriminator, as in traditional GANs. Additionally, InfoGANs introduce an auxiliary loss that encourages the discriminator to estimate the latent codes accurately. This loss is typically based on the mutual information between the latent codes and the features extracted from the discriminator's intermediate layers. By maximizing the mutual information between the latent codes and the generated data, InfoGANs promote the learning of disentangled representations. Disentangled representations imply that each latent code corresponds to a specific attribute or feature, allowing for systematic control over the generated data. For example, in a face generation task, latent codes can encode attributes such as pose, hair color, or facial expression, enabling the generation of diverse and controllable variations of faces [66, 67].

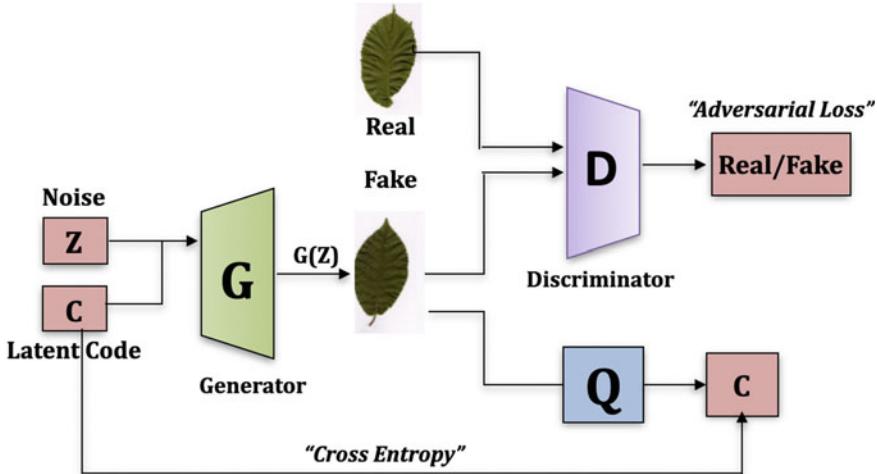


Fig. 13 The framework of Information Maximizing Generative Adversarial Networks (Info-GANs), here Q:Q-network to learn disentangled representation

3.8 Super-Resolution Generative Adversarial Networks (SRGANs)

Super-resolution Generative Adversarial Networks (SRGANs) [68] are designed to enhance the resolution and details of low-resolution images by generating its high-resolution counterparts. Unlike traditional image super-resolution methods that rely on handcrafted features or optimization-based techniques, SRGANs utilize the power of deep learning to automatically learn the mapping from low to high resolution [69].

The architecture of SRGANs typically consists of two main components: a generator network and a discriminator network. On the other hand, the discriminator network is responsible for distinguishing between real high-resolution images and generated high-resolution images. It employs a series of convolutional layers to upsample the input image and refine its details, gradually improving the resolution and quality of the generated image. The discriminator network, on the other hand, is responsible for distinguishing between real high-resolution images and generated high-resolution images. To further enhance the visual quality of the generated images, SRGANs often incorporate additional loss functions [70]. One common loss function is the content loss, which measures the perceptual similarity between the generated and real high-resolution images using feature representations extracted from pre-trained deep neural networks [68]. Another loss function is the pixel-wise mean squared error (MSE) loss, which encourages the generator to produce high-resolution images that closely resemble the ground truth images at the pixel level [71].

SRGANs offers a data-driven and an end-to-end solution for enhancing the resolution and details of low-resolution images, providing valuable tools for applications that require high-quality image reconstruction and restoration.

3.9 Generative Image Inpainting

The primary objective of Generative Image Inpainting is to learn a model that can understand the context and structure of the surrounding image regions and use this information to generate plausible content for the missing regions [72].

Generative Image Inpainting with GANs involves two key components: a generator network and a discriminator network. The generator network takes as input an incomplete or damaged image and generates a completed version of the image, filling in the missing regions [73]. On the other hand, the discriminator network tries to distinguish between the generated inpainted images and real complete images. In addition to the adversarial loss, GAN-based inpainting models often employ additional losses to ensure the quality and coherence of the inpaintings [74]. These losses include perceptual loss, which measures the similarity between the inpainted and ground truth images at a higher level by comparing their feature representations extracted from pre-trained networks. And the content loss, which measures the pixel-wise difference between the inpainted and ground truth images [72].

To guide the inpainting process and ensure consistency with the surrounding content, GAN based inpainting models may utilize techniques such as contextual attention or partial convolutions. Contextual attention mechanisms allow the generator to attend to relevant regions in the input image that guides the inpainting process [72]. Partial convolutions adaptively mask out the missing regions during the convolutional operations, preserving the integrity of the surrounding content [75].

Generative Image Inpainting with GANs have proven effective in various applications including image restoration, object removal, and image editing. It offers an automated and data-driven approach to fill in the missing or damaged regions, leveraging the power of adversarial training to generate realistic and visually coherent inpaintings.

4 Applications of Generative AI in Plant Phenomics

In recent years, there has been a growing body of research focusing on the applications of Generative AI in plant phenomics. This is mainly due to the advancement of GAN based architectures (previous section). In this chapter, we categorise these applications based on various plant phenotyping tasks namely, (a) Leaf segmentation and counting, (b) Disease segmentation and classification, (c) Weed recognition and (d) Root segmentation (Fig. 14).

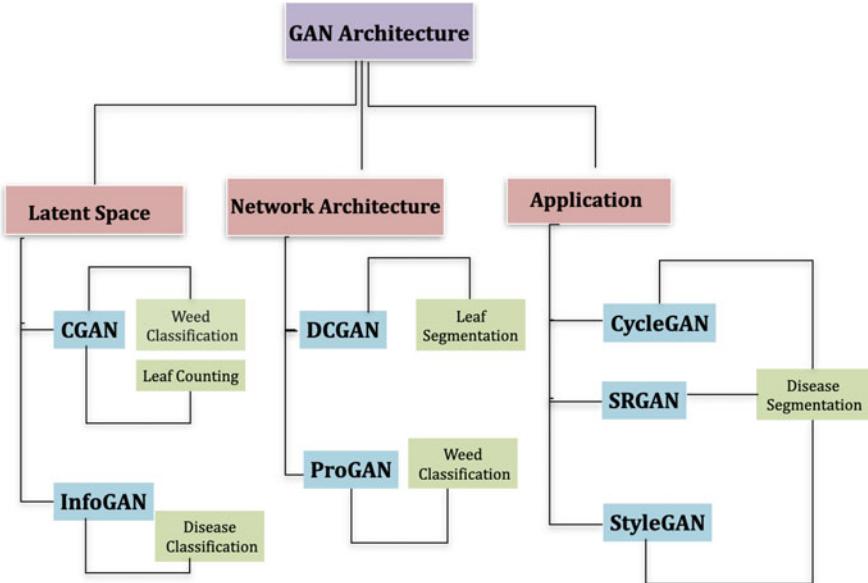


Fig. 14 An overview of GANs variants and applications

4.1 Leaf Segmentation and Counting

Based on the CVPPP dataset [15], the authors in [76] employed Conditional GAN (cGAN) with condition on the number of leaves of the plant canopy. The authors termed their plant canopy generation framework as *ARIGAN* and showed the utilisation of generated dataset on leaf counting task. Similarly, Purbaya et al. [77] introduced leaf shape constraints to generate plant canopy images. It is to be noted that both studies accurately synthesized leaf shapes but were limited with respect to leaf texture and background generation. Another study [78] employed Conditional GAN (cGAN) with leaf segmentation masks as the conditional constraint. Based on the CVPPP dataset [15], the authors showed the improvement on leaf counting task using Mask R-CNN [79]. cGAN was also employed as a generative growth model trained on aligned image pairs i.e., pairs with plant image at early growth stage and its later growth stage in [80]. This training paradigm permits temporal (growth) prediction of plants.

In another study [81], authors employed StackGAN to generate synthetic plant images. The authors termed their GAN based framework as *CropPainter*. In contrast to the aforementioned studies where the conditional image synthesis is based on a single label describing the image, theirs' conditional image synthesis is dependent on a feature vector describing the phenotypic data of the image. It is to be noted that due to the plant species gap, deep learning models trained on a specific plant dataset is not generalisable to other plant species. To overcome this limitation, authors in [82]

attempted to this reduce domain shift based on adversarial learning. Specifically, the authors proposed a GAN based framework for leaf counting on new dataset (target domain) with the knowledge learnt on source domain dataset, requiring no additional annotations for the target domain.

In plant phenomics, various studies [83–85] highlighted the difference in the spectral signatures of plant leaves from the non-plant leaves region (such as soil, weed pot, etc.) in multi-spectral domain such as chlorophyll fluorescence, near infrared (NIR), thermal, hyperspectral etc. in contrast to visible image. This facilitates leaf segmentation and counting via traditional segmentation algorithms. However, the acquisition of multi-spectral images is highly expensive and requires special imaging protocols. In this context, various GAN based frameworks have been proposed for multi-modal translation. This not only facilitates the aforementioned phenotyping tasks but also the generation of multi-modal images from widely available visible images. For example, authors in [86] employed cGAN architecture for image to image translation i.e., generating NIR reflectance images from the visible spectrum of UAV field images. These generated images were then employed for leaf segmentation task. Recent study [87] proposed an attention based Pix2Pix GAN to generate NIR images from the corresponding visible images. The authors also improved the resolution of generated images for accurate plant segmentation.

4.2 Disease Classification and Segmentation

Authors in [88] presented label constraint DCGAN for augmentation of tea leaf dataset exhibiting different classes of disease symptoms. The label constraint denoted the disease class and the generated dataset was employed to improve disease classification accuracy. This study also highlighted the improvement of the classification model trained on GAN augmented dataset compared to basic augmented dataset. Similarly, another study [89] employed cGAN to synthesise tomato disease leaf images based on publicly available PlantVillage dataset [8] and showed the improvement on disease classification based on this augmented dataset. In contrast, DCGAN was employed in [90, 91] to generate five disease classes of tomato leaf images trained on PlantVillage dataset. However, there is no empirical evidence yet to contrast the performance of cGAN and DCGAN on tomato dataset. Another study [92], proposed to employ DCGAN to generate seven different leaf disease classes. This study highlights the efficacy of GAN models to learn the inter-class and intra-class distribution of disease symptoms.

A comparative analysis was conducted in [93] with vanilla GAN and DCGAN to generate tea leaf images for four disease classes. It was empirically observed that the classifiers trained on augmented data generated with DCGAN showed better accuracy compared to vanilla GAN. Authors in [94, 95] adapted DCGAN for generating citrus disease images based on PlantVillage dataset to improve the disease classification accuracy. To control the shape of disease regions in leaf images, authors [96] proposed a binary generator network. The generated images was subsequently employed in

an edge-smoothing algorithm to synthesize novel leaf disease dataset. Similar to the previous studies, GAN augmented data showed improvement in classification accuracy with AlexNet. Instead of generating disease lesion spots with the generator, Faster R-CNN was employed in [97] to extract disease spots which were subsequently employed in DCGAN for data augmentation. DCGAN has also been employed to generate infrared (IR) images of apple leaf disease images and the augmented dataset was used for disease spot segmentation via SegNet [98].

Cycle-consistent adversarial network (CycleGAN) was proposed in [99] to augment apple leaf disease dataset, specifically the model transformed healthy leaf images to disease images. The authors combined the GAN augmented dataset with traditional augmented dataset for improving disease detection implemented using YOLO. Similarly, authors in [100] employed CycleGAN for generating grape leaf disease dataset to improve the disease classification task. In contrast to vanilla CycleGAN, U-Net based CycleGAN was employed in [101] for tomato leaf dataset with nine different disease classes. A comparative analysis highlighted the efficacy of the latter to synthesise texture level details compared to the former. Another attempt to improve CycleGAN was based on the modification of the vanilla CycleGANs' discriminator with one label output as disease classification label and other label output as real/fake based on the generated image [102]. Similarly, the perceptual quality of the generated leaf disease images was improved with the inclusion of reconstruction loss in CycleGAN in [19]. The proposed framework was termed as *AR-GAN* (activation reconstruction GAN) and was also employed in [103] for cucumber leaf disease dataset. To further expand the variance in the augmented dataset, leaf segmentation module was included in vanilla CycleGAN [104]. This module allowed the generator to focus on leaf regions for synthesising disease spots and neglect the background regions. The authors termed the proposed CycleGAN based framework as *Leaf-GAN*.

Based on the publicly available PlantVillage dataset, authors in [105] employed another GAN based framework StyleGAN and generated leaf images with 42 disease classes. The GAN augmented dataset was combined with traditional data augmentation methods for disease classification task. In another study, the authors [106] proposed to utilise style consistent image translation GAN termed as *SCITGAN* for synthesising leaf images belonging to five disease classes. Authors in [47] adapted Style-Generative Adversarial Network Adaptive Discriminator Augmentation (SG2-ADA) with a laplacian filter to generate rice leaf disease images. The generated images were employed to improve disease detection based on Faster Region Based Convolutional Neural Network (faster-RCNN) and Single Shot Detector (SSD).

It is evident from the aforementioned discussion that for disease classification, the majority of the studies used publicly available PlantVillage dataset. However, this dataset is limited in resolution and affects the data generation module and subsequently the disease classification and segmentation task. To overcome this limitation, SRGAN was utilised as a preprocessing step in [107]. Specifically, the original dataset is downsampled to first generate the training data for SRGAN. The super-resolved images were then employed to perform disease classification showing an improved performance compared to the original low-resolution dataset. In another

study [108], Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN) was employed to synthesise high-resolution images of tomato leaf disease dataset (PlantVillage dataset). Similar findings were observed with respect to the improvement in classification accuracy. However, the authors did not conduct a comparative analysis of different super-resolution frameworks necessary for reasonable improvement in classification accuracy with minimum computational cost.

With respect to disease segmentation, the task is implemented in a supervised paradigm with training on GAN augmented data and labels. In contrast to this strategy, few studies employed GAN for an end-to-end disease segmentation. For instance, authors in [109] proposed to utilise Pix2Pix GAN for reconstructing the diseased leaf regions into its healthy counterparts. The difference between the disease image and the generated image is utilised for disease segmentation. This study only focused on the colour reconstructability of the disease region and it fails in scenarios with textural changes of the symptomatic regions. To overcome this limitation, a recent study [110] presented a contextual attention based GAN inpainting network that learns both color and textural reconstruction of disease regions. They also incorporated GAN based contextual attention maps to improve the disease segmentation.

4.3 Weed Recognition

Authors in [111] adapted the cGAN architecture with spatially-adaptive normalization to generate images conditioned on the mask denoting the plant region, leaving the background region unchanged. This resulted in the generation of semi-artificial images with focus only on the plant regions. Experimental results in this study showed the improvement on weed segmentation task when trained with GAN augmented dataset. In another study [112], authors employed DCGAN to generate synthetic tomato dataset for weed recognition task. They conducted an ablation to obtain an efficient DCGAN model based on FID metric [113] (FID quantifies the quality and diversity of the generated images).

The previously mentioned GAN based frameworks were presented for visible images consisting of only one plant canopy. For high-throughput screening of field images, authors in [114] employed DCGAN for segmentation of multi-spectral UAV images (weedNet). In contrast to vanilla GAN, the discriminator in this study utilised three inputs i.e., generated data, labelled and unlabelled data to generate confidence maps for each class (i.e soil, plant, weed etc.) and a binary label (real/fake). Authors in [115] also utilised a similar semi-supervised GAN model for weed classification based only on RGB images. The study conducted extensive comparisons with supervised methods to demonstrate the efficiency of the proposed training paradigm with limited training data.

4.4 Root Segmentation

Due to thin structures, root segmentation usually results in small gaps/holes that generate erroneous results with respect to root system architecture [116–118]. Few studies in plant phenomics employed GAN based frameworks for recovering these gaps. This problem is formulated as image inpainting. For example, authors in [119, 120] proposed to inpaint local gap patterns and a global discriminator that adopts policy gradient with an inpainting generator. In contrast to local patch inpainting, global discriminator utilises global root information for accurate inpainting. These studies highlight the application of GANs as a post-processing step in improving the segmentation task.

5 Evaluation Metrics

As evident from the previous section, various state-of-the-art GAN based frameworks have been utilised for generating novel plant phenotypic data such as Arabidopsis plant canopy dataset [78], tomato leaf disease dataset [19, 106], weed dataset [112] to name a few. The objective analysis of the generated image data for plant phenotyping is implemented based on (a) Domain-specific metrics and (b) Similarity metrics [121]. The former category of metrics are employed to quantify the performance of deep learning models on specific plant phenotyping tasks trained on GAN augmented datasets. The latter quantifies the similarity between the generated image dataset and original image dataset. In the subsequent subsections, we elucidate these metrics.

5.1 Domain-Specific Metrics

Domain-specific metrics are used for evaluating the performance of deep learning models trained with both GAN augmented and original plant image datasets. The metrics highlight the improvement of these models in contrast to the models trained only with original plant image datasets. Here, deep learning models typically include classification, segmentation networks etc. employed for specific plant phenotyping tasks. We enumerate the commonly used metrics below:

- Precision [122]: This metric is commonly employed for phenotyping tasks such as disease classification, disease segmentation etc. Precision measures the accuracy of positive predictions for a particular class. For example, it is computed as the ratio of correctly classified rust disease class and predicted rust disease class in a disease classification task. In the context of disease segmentation task, it is computed as the ratio of correctly classified disease pixels to the predicted disease pixels.
- Recall [122]: This metric is employed for phenotyping tasks such as disease classification, disease segmentation etc. Recall measures the completeness of positive

predictions for a particular class. For example, it is computed as the ratio of correctly classified rust disease class out of all the actual rust disease class. In the context of disease segmentation task, it is computed as the ratio of correctly classified disease pixels to the actual disease pixels.

- F1-Score [122]: F1-score combines precision and recall, providing a balanced measure of the overall accuracy of deep learning models.
- Intersection-Over-Union (Jaccard index) [79]: Intersection-Over-Union (IoU) is commonly employed for phenotyping tasks such as disease segmentation, plant canopy segmentation etc. This metric is computed as the overlap between the predicted and the ground truth mask. It measures the efficiency of the model to separate the phenotypic traits such as disease symptoms, plant leaves etc. from the background in the plant image data.
- Dice Coefficient (Dice) [79]: This metric is commonly employed for phenotyping tasks such as disease segmentation, plant canopy segmentation etc. It is computed as a harmonic mean of precision and recall. Both IoU and Dice penalise false positive predictions, but IoU penalises under and over-segmentation more than Dice.
- Symmetric Best Dice (SBD) [123]: This metric is commonly employed for phenotyping tasks such as multi-instance segmentation of leaves, spikes, grains etc. Firstly, the maximum Dice for each phenotypic instance is calculated and the average of these Dice is then computed, referred to as SBD metric.

5.2 *Similarity Metrics*

This category of metrics are employed to compare the proximity of the generated plant image datasets with the original plant image datasets. We enumerate the commonly used metrics below:

- Frechet Inception Distance (FID) [113]: FID measures the similarity between the distribution of original images and generated images based on the feature representations of an inception network. It quantifies the quality and diversity of the generated images. Lower FID values signifies better quality.
- Frechet ResNet Distance (FRD) [124]: FRD is an extension of the Fréchet Inception Distance (FID) metric. It employs pre-trained ResNet [125] to extract feature representations from both original and generated image data. Specifically, FRD measures the dissimilarity between two distributions using the Fréchet distance, considering both the mean and covariance of the feature representations. Lower FRD values signifies better quality.
- Neural image assessment (NIMA) [126]: NIMA evaluates aesthetic quality and perceptual significance of the generated images.
- Inception Score[127]: Inception Score measures the diversity and quality of generated images by assessing the conditional label entropy and image quality. It considers both diversity and clarity of the generated phenotypic data.

- GAN-Train [97, 102]: GAN-Train refers to training deep learning models with plant images generated by GANs and then computing the accuracy on original plant images. Based on the plant phenotyping task, the training models can be a disease classification network and its performance quantified using classification metrics or it can be an instance segmentation network evaluated with segmentation metrics (previous subsection). This metric reflects the diversity and authenticity of GAN augmented data.
- GAN-Test [97, 102]: In contrast to the previous metric, GAN-Test refers to training deep learning models with original plant images and images generated via GANs are used to compute its accuracy. This metric signifies the closeness of the GAN augmented images to original images.
- Generalizability Test: Various studies in plant phenotyping [104, 106] evaluated the generality of deep learning models on unseen plant image data based on different training paradigms. These different training paradigms include (1) training with images from original plant datasets, (2) training with images from original plant images and basic augmented plant images and (3) training using images from both original and generated plant datasets. As mentioned previously, these deep learning models can belong to classification networks for leaf disease dataset, segmentation networks for plant canopy dataset etc. with their corresponding accuracy metrics.

In addition to these objective metrics, subjective analysis have also been conducted based on the generated images. For example, the authors in [97, 102] compared training and validation learning curves of deep learning models trained on (a) standard augmented datasets and (b) GAN augmented datasets. This subjective analysis highlights the improvement of model efficacy trained with GAN augmented dataset. Few studies [102, 128] subjectively analysed the performance of GAN frameworks based on the deep feature visualisation of the generated images. For example, deep features of augmented plant images and original plant images are extracted from VGG, EfficientNet etc. [128] and visualised using t-distributed stochastic neighbour embedding (t-SNE) [129]. The overlap in t-SNE visualisation denotes that the generated image reflects the original plant image data. Similarly, the difference in the deep features of the augmented images belonging to different phenotypic traits (such as different disease classes) can also be visualised using t-SNE. With distribution far and wide meaning the generated images capture the subtle features of original plant image data.

The evaluation metrics discussed in the previous subsections provide a comprehensive understanding of how to quantitatively assess the quality, accuracy, diversity, and perceptual aspects of generated plant phenotypic data. Considering the specific requirements of plant phenotyping task, selecting appropriate evaluation metrics will help researchers in effectively evaluating and improving generative AI models.

5.3 *Limitations and Scope of Generative AI in Plant Phenomics*

While Generative AI has shown great promise in the field of plant phenomics, it is important to acknowledge its current limitations, which serve as opportunities for future advancements. Here are some key limitations and potential avenues for improvement:

- The performance of Generative AI models heavily relies on the availability of high-quality, diverse, and well-annotated training data. However, obtaining large-scale and accurately labelled datasets can be challenging due to the large biological variability of plant phenotypic data. This demands innovations with respect to training objectives for GAN with limited dataset.

The advancement in formulating GAN based dataset generation as joint multimodal learning that simultaneously synthesizes images belonging to different phenotypic data i.e., data for disease classification and segmentation. Another research investigation includes training GAN in a semi-supervised paradigm that often requires limited data.

While Generative AI models trained on specific datasets can generate impressive results, there is a need for improved transferability and generalization capabilities. Models that can generalize across different plant species, growth conditions, and experimental setups would be more versatile and applicable in various research and agricultural settings. Future research should focus on developing transfer learning techniques and domain adaptation strategies to enhance model performance in diverse scenarios.

- Generative AI models are often considered as black boxes due to their complex architectures. Understanding the decision-making process of these models is crucial for their practical applications. Researchers should invest in developing methods to improve the interpretability and explainability of Generative AI models in the context of plant phenomics. This would enable researchers to have a better understanding of the underlying biological mechanisms and make informed decisions.

It is evident from the application of GANs in plant phenomics, metrics to assess the quality of the generated images vary greatly even for similar phenotypic datasets. This limits the evaluation of different GANs models for comprehensive understanding and future advancements.

- Plant phenomics encompasses a wide range of biological, physiological, and ecological factors. Integrating domain knowledge into Generative AI models can enhance their performance and may result in biologically meaningful generated outputs. Future applications should aim to incorporate domain-specific knowledge, such as genetic information, plant physiology, and environmental interactions, to design more robust and accurate Generative AI models.
- Many generative AI models require significant computational resources and time for training and inference. This poses challenges in scaling up these models to

handle large-scale phenotypic datasets and real-time applications. Future developments should focus on optimizing model architectures, designing efficient training algorithms, and utilizing parallel computing techniques to improve scalability and computational efficiency.

As researchers continue to address these limitations, the future applications of Generative AI in plant phenomics hold great promise. Overcoming these challenges will enable researchers to unlock the full potential of Generative AI models, leading to breakthroughs in crop breeding, adaptive management, disease detection, and sustainable agriculture.

6 Conclusion

In this chapter, we surveyed various image augmentation methods employed in plant primarily focusing on GAN based methods. The core concepts of various GAN variants employed in plant phenomics have been detailed. This helps researchers to select suitable GAN frameworks based on the quantity and quality of available phenotypic data. Since, various metrics have been proposed to assess the perceptual quality of GAN augmented datasets, in this chapter we also enumerate commonly adopted metrics for future standardisation. Furthermore, we discussed the current application of Generative AI in plant phenomics and possible avenues for future research. To summarise, this chapter will provide an enhanced understanding of GAN based augmentations and encourage plant phenomics research.. With ongoing advancements and interdisciplinary collaborations, Generative AI will play an increasingly significant role in revolutionizing plant phenomics and shaping the future of agriculture.

References

1. Fiorani, F., & Schurr, U. (2013). Future scenarios for plant phenotyping. *Annual Review of Plant Biology*, 64, 267–291.
2. Fasoula, D. A., Ioannides, I. M., & Omirou, M. (2020). Phenotyping and plant breeding: Overcoming the barriers. *Frontiers in Plant Science*, 10, 1713.
3. Li, L., Zhang, Q., & Huang, D. (2014). A review of imaging techniques for plant phenotyping. *Sensors* 14(11), 20 078–20 111.
4. Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145, 311–318.
5. Esgario, J. G., Krohling, R. A., & Ventura, J. A. (2020). Deep learning for classification and severity estimation of coffee leaf biotic stress. *Computers and Electronics in Agriculture*, 169, 105162.
6. Söderkvist, O. (2001). Computer vision classification of leaves from swedish trees.
7. Ubbens, J. R., & Stavness, I. (2017). Deep plant phenomics: A deep learning platform for complex plant phenotyping tasks. *Frontiers in Plant Science*, 8, 1190.
8. Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7, 1419.

9. David, E., Madec, S., Sadeghi-Tehran, P., Aasen, H., Zheng, B., Liu, S., Kirchgessner, N., Ishikawa, G., Nagasawa, K., Badhon, M. A. et al. (2020). Global wheat head detection (gwhd) dataset: A large and diverse dataset of high-resolution rgb-labelled images to develop and benchmark wheat head detection methods. *Plant Phenomics*.
10. Sordo, M., & Zeng, Q. (2005). On sample size and classification accuracy: A performance comparison. In *Biological and medical data analysis: 6th International symposium, ISBMDA*. Proceedings 6 (pp. 193–201). Springer.
11. Prusa, J., Khoshgoftaar, T. M., & Seliyi, N. (2015). The effect of dataset size on training tweet sentiment classifiers. In *2015 IEEE 14th International conference on machine learning and applications ICMLA* (pp. 96–102). IEEE.
12. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition* (pp. 248–255). IEEE.
13. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer vision-ECCV, 13th European conference proceedings, Part V 13* (pp. 740–755). Springer.
14. Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88, 303–338.
15. Minervini, M., Fischbach, A., Scharr, H., & Tsaftaris, S. A. (2016). Finely-grained annotated datasets for image-based plant phenotyping. *Pattern Recognition Letters*, 81, 80–89.
16. Uchiyama, H., Sakurai, S., Mishima, M., Arita, D., Okayasu, T., Shimada, A., & Taniguchi, R.-I. (2017). An easy-to-setup 3d phenotyping platform for komatsuna dataset. In *Proceedings of the IEEE international conference on computer vision workshops* (pp. 2038–2045).
17. Shadrin, D. G., Kulikov, V., & Fedorov, M. V. (2018). Instance segmentation for assessment of plant growth dynamics in artificial soilless conditions. *BMVC*, p. 329.
18. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139–144.
19. Nazki, H., Yoon, S., Fuentes, A., & Park, D. S. (2020). Unsupervised image translation using adversarial networks for improved plant disease recognition. *Computers and Electronics in Agriculture*, 168, 105117.
20. Fahlgren, N., Gehan, M. A., & Baxter, I. (2015). Lights, camera, action: High-throughput plant phenotyping is ready for a close-up. *Current Opinion in Plant Biology*, 24, 93–99.
21. Ren, M., & Zemel, R. S. (2017). End-to-end instance segmentation with recurrent attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6656–6664).
22. Salvador, A., Bellver, M., Campos, V., Baradad, M., Marques, F., Torres, J., & Giro-i Nieto, X. (2017). Recurrent neural networks for semantic instance segmentation. *arXiv preprint arXiv:1712.00617*.
23. Giuffrida, M. V., Doerner, P., & Tsaftaris, S. A. (2018). Pheno-deep counter: A unified and versatile deep learning architecture for leaf counting. *The Plant Journal*, 96(4), 880–890.
24. Cruz, A. C., Luvisi, A., De Bellis, L., & Ampatzidis, Y. (2017). Vision-based plant disease detection system using transfer and deep learning. In *Asabe annual international meeting. American Society of Agricultural and Biological Engineers*, (p. 1).
25. DeChant, C., Wiesner-Hanks, T., Chen, S., Stewart, E. L., Yosinski, J., Gore, M. A., Nelson, R. J., & Lipson, H. (2017). Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. *Phytopathology*, 107(11), 1426–1432.
26. Dwibedi, D., Misra, I., & Hebert, M. (2017). Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 1301–1310).
27. Ubbens, J., Cieslak, M., Prusinkiewicz, P., & Stavness, I. (2018). The use of plant models in deep learning: An application to leaf counting in rosette plants. *Plant Methods*, 14, 1–10.
28. Shapiro, L. G., & Stockman, G. C. (2001). *Computer vision*. Pearson.
29. Gonzalez, R. C. (2009). *Digital image processing*. Pearson education India.

30. Afifi, M., Price, B., Cohen, S., & Brown, M. S. (2019). When color constancy goes wrong: Correcting improperly white-balanced images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1535–1544).
31. Taylor, L., & Nitschke, G. (2018). Improving deep learning with generic data augmentation. In *IEEE symposium series on computational intelligence (SSCI), IEEE* (pp. 1542–1547).
32. da Costa, G. B. P., Contato, W. A., Nazare, T. S., Neto, J. E., & Ponti, M. (2016). An empirical study on the effects of different types of noise in image classification tasks. arXiv preprint [arXiv:1609.02781](https://arxiv.org/abs/1609.02781).
33. Tang, Y., & Eliasmith, C. (2010). Deep networks for robust visual recognition. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 1055–1062).
34. Nazaré, T. S., da Costa, G. B. P., Contato, W. A., & Ponti, M. (2018). Deep convolutional neural networks and noisy images. In *Progress in pattern recognition, image analysis, computer vision, and applications: 22nd Iberoamerican Congress, CIARP, Proceedings 22* (pp. 416–424). Springer.
35. Kuznichov, D., Zvirin, A., Honen, Y., & Kimmel R. (2019). Data augmentation for leaf segmentation and counting tasks in rosette plants. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*.
36. Toda, Y., Okura, F., Ito, J., Okada, S., Kinoshita, T., Tsuji, H., & Saisho, D. (2019). Learning from synthetic dataset for crop seed instance segmentation. *BioRxiv* (p. 866921).
37. Gomes, D. P. S., & Zheng, L. (2020). Recent data augmentation strategies for deep learning in plant phenotyping and their significance. In *Digital image computing: Techniques and applications (DICTA)* (pp. 1–8). IEEE.
38. Qiongyan, L., Cai, J., Berger, B., Okamoto, M., & Miklavcic, S. J. (2017). Detecting spikes of wheat plants using neural networks with laws texture energy. *Plant Methods*, 13, 1–13.
39. Aristid, L. (1968). Mathematical models for cellular interactions in development ii. Simple and branching filaments with two-sided inputs. *Journal of Theoretical Biology*, 18(3), 300–315.
40. Allen, M., DeJong, T., & Prusinkiewicz, P. (2004). L-peach, an l-systems based model for simulating the architecture and carbon partitioning of growing fruit trees. *VII International Symposium on Modelling in Fruit Research and Orchard Management*, 707, 71–76.
41. Leitner, D., Klepsch, S., Knieß, A., & Schnepf, A. (2010). The algorithmic beauty of plant roots—an l-system model for dynamic root growth simulation. *Mathematical and Computer Modelling of Dynamical Systems*, 16(6), 575–587.
42. Cieslak, M., Khan, N., Ferraro, P., Soolanayakanahally, R., Robinson, S. J., Parkin, I., McQuillan, I., & Prusinkiewicz, P. (2022). L-system models for image-based phenomics: Case studies of maize and canola. *In Silico Plants*, 4(1), diab039.
43. Mundermann, L., Erasmus, Y., Lane, B., Coen, E., & Prusinkiewicz, P. (2005). Quantitative modeling of arabidopsis development. *Plant Physiology*, 139(2), 960–968.
44. Jallas, E., Sequeira, R., Martin, P., Turner, S., & Papajorgji, P. (2009). Mechanistic virtual modeling: Coupling a plant simulation model with a three-dimensional plant architecture component. *Environmental Modeling and Assessment*, 14, 29–45.
45. España, M. L., Baret, F., Aries, F., Chelle, M., Andrieu, B., & Prévot, L. (1999). Modeling maize canopy 3d architecture: Application to reflectance simulation. *Ecological Modelling*, 122(1–2), 25–43.
46. Ward, D., & Moghadam, P. (2020). Scalable learning for bridging the species gap in image-based plant phenotyping. *Computer Vision and Image Understanding*, 197, 103009.
47. Haruna, Y., Qin, S., & Mbyamm Kiki, M. J. (2023). An improved approach to detection of rice leaf disease with gan-based data augmentation pipeline. *Applied Sciences*, 13(3), 1346.
48. Goodfellow, I. (2016). Nips 2016 tutorial: Generative adversarial networks. arXiv preprint, [arXiv:1701.00160](https://arxiv.org/abs/1701.00160).
49. Radford, A., Metz, L., & Chintala, S. (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint, [arXiv:1511.06434](https://arxiv.org/abs/1511.06434).
50. Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint, [arXiv:1411.1784](https://arxiv.org/abs/1411.1784).

51. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. (2016). Generative adversarial text to image synthesis. In *International conference on machine learning* (pp. 1060–1069). PMLR.
52. Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125–1134).
53. O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. [arXiv:1511.08458](https://arxiv.org/abs/1511.08458).
54. Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431–3440).
55. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
56. Yan, L. C., Yoshua, B., & Geoffrey, H. (2015). Deep learning. *Nature*, 521(7553), 436–444.
57. Zhu, J.-Y., Park, T., Isola, & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223–2232).
58. Yi, Z., Zhang, H., Tan, P., & Gong, M. (2017). Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE international conference on computer vision* (pp. 2849–2857).
59. Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. [arXiv:1710.10196](https://arxiv.org/abs/1710.10196).
60. Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., & Hadsell, R. (2016). Progressive neural networks. [arXiv:1606.04671](https://arxiv.org/abs/1606.04671).
61. Durall, R., Chatzimichailidis, A., Labus, P., & Keuper, J. (2020). Combating mode collapse in gan training: An empirical analysis using hessian eigenvalues. [arXiv:2012.09673](https://arxiv.org/abs/2012.09673).
62. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of wasserstein gans. *Advances in Neural Information Processing Systems* 30.
63. Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4401–4410).
64. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8110–8119).
65. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in Neural Information Processing Systems* 29.
66. Kurutach, T., Tamar, A., Yang, G., Russell, S. J., & Abbeel, P. (2018). Learning plannable representations with causal infogan. *Advances in Neural Information Processing Systems* 31.
67. Spurr, A., Aksan, E., & Hilliges, O. (2017). Guiding infogan with semi-supervision. In *Machine learning and knowledge discovery in databases: European conference, ECML PKDD, Proceedings, Part I* (pp. 119–134). Springer.
68. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4681–4690).
69. Nasrollahi, K., & Moeslund, T. B. (2014). Super-resolution: A comprehensive survey. *Machine Vision and Applications*, 25, 1423–1468.
70. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., & Change Loy, C. (2018). Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*.
71. Deng, X. (2018). Enhancing image quality via style transfer for single image super-resolution. *IEEE Signal Processing Letters*, 25(4), 571–575.
72. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., & Huang, T. S. (2018). Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5505–5514).

73. Iizuka, S., Simo-Serra, E., & Ishikawa, H. (2017). Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4), 1–14.
74. Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., & Li, H. (2017). High-resolution image inpainting using multi-scale neural patch synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6721–6729).
75. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., & Huang, T. S. (2019). Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 4471–4480).
76. Valerio Giuffrida, M., Scharr, H., & Tsaftaris, S. A. (2017). Arigan: Synthetic arabidopsis plants using generative adversarial network. In *Proceedings of the IEEE international conference on computer vision workshops* (pp. 2064–2071).
77. Purbaya, M. E., Setiawan, N. A., & Adjji, T. B. (2018). Leaves image synthesis using generative adversarial networks with regularization improvement. In *2018 International conference on information and communications technology (ICOIACT)* (pp. 360–365). IEEE.
78. Zhu, Y., Aoun, M., Krijn, M., Vanschoren, J., & Campus, H. T. (2018). Data augmentation using conditional generative adversarial networks for leaf counting in arabidopsis plants. In *BMVC*, p. 324.
79. He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961–2969).
80. Drees, L., Junker-Frohn, L. V., Kierdorf, J., & Roscher, R. (2021). Temporal prediction and evaluation of brassica growth in the field using conditional generative adversarial networks. *Computers and Electronics in Agriculture*, 190, 106415.
81. Duan, L., Wang, Z., Chen, H., Fu, J., Wei, H., Geng, Z., & Yang, W. (2022). Croppainter: An effective and precise tool for trait-to-image crop visualization based on generative adversarial networks. *Plant Methods*, 18(1), 1–11.
82. Valerio Giuffrida, M., Dobrescu, A., Doerner, P., & Tsaftaris, S. A. (2019). Leaf counting without annotations using adversarial unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*.
83. Henke, M., Junker, A., Neumann, K., Altmann, T., & Gladilin, E. (2019). Comparison of feature point detectors for multimodal image registration in plant phenotyping. *Plos One*, 14(9), e0221203.
84. Henke, M., Junker, A., Neumann, K., Altmann, T., & Gladilin, E. (2020). A two-step registration-classification approach to automated segmentation of multimodal images for high-throughput greenhouse plant phenotyping. *Plant Methods*, 16(1), 95.
85. Sapoukhina, N., Samiei, S., Rasti, P., & Rousseau, D. (2019). Data augmentation from rgb to chlorophyll fluorescence imaging application to leaf segmentation of arabidopsis thaliana from top view images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*.
86. Aslahishahri, M., Stanley, K. G., Duddu, H., Shirtliffe, S., Vail, S., Bett, K., Pozniak, C., & Stavness, I. (2021). From rgb to nir: Predicting of near infrared reflectance from visible spectrum aerial images of crops. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1312–1322).
87. Shukla, A., Upadhyay, A., Sharma, M., Chinnusamy, V., & Kumar, S. (2022). High-resolution nir prediction from rgb images: Application to plant phenotyping. In *2022 IEEE international conference on image processing (ICIP)* (pp. 4058–4062). IEEE.
88. Hu, G., Wu, H., Zhang, Y., & Wan, M. (2019). A low shot learning method for tea leaf's disease identification. *Computers and Electronics in Agriculture*, 163, 104852.
89. Abbas, A., Jain, S., Gour, M., & Vankudothu, S. (2021). Tomato plant disease detection using transfer learning with c-gan synthetic images. *Computers and Electronics in Agriculture*, 187, 106279.
90. Wu, Q., Chen, Y., & Meng, J. (2020). Dcgan-based data augmentation for tomato leaf disease identification. *IEEE Access* 8 , 98 716–98 728.
91. Gomaa, A. A., & Abd El-Latif, Y. M. (2021). Early prediction of plant diseases using cnn and gans. *International Journal of Advanced Computer Science and Applications* 12(5).

92. Hu, W.-J., Xie, T.-Y., Li, B.-S., Du, Y.-X., & Xiong, N. N. (2021). An edge intelligence-based generative data augmentation system for iot image recognition tasks. *Journal of Internet Technology*, 22(4), 765–778.
93. Yuwana, R. S., Fauziah, F., Heryana, A., Krisnandi, D., Kusumo, R. B. S., & Pardede, H. F. (2020). Data augmentation using adversarial networks for tea diseases detection. *Jurnal Elektronika dan Telekomunikasi*, 20(1), 29–35.
94. Lan, L., You, L., Zhang, Z., Fan, Z., Zhao, W., Zeng, N., Chen, Y., & Zhou, X. (2020). Generative adversarial networks and its applications in biomedical informatics. *Frontiers in Public Health*, 8, 164.
95. Zhang, M., Liu, S., Yang, F., & Liu, J. (2019). Classification of canker on small datasets using improved deep convolutional generative adversarial networks, *IEEE Access* 7, 49 680–49 690.
96. Sun, R., Zhang, M., Yang, K., & Liu, J. (2020). Data enhancement for plant disease classification using generated lesions. *Applied Sciences*, 10(2), 466.
97. Chen, Y., & Wu, Q. (2023). Grape leaf disease identification with sparse data via generative adversarial networks and convolutional neural networks. *Precision Agriculture*, 24(1), 235–253.
98. Douarre, C., Crispim-Junior, C. F., Gelibert, A., Tougne, L., & Rousseau, D. (2019). Novel data augmentation strategies to boost supervised segmentation of plant disease. *Computers and Electronics in Agriculture*, 165, 104967.
99. Tian, Y., Yang, G., Wang, Z., Li, E., & Liang, Z. (2019). Detection of apple lesions in orchards based on deep learning methods of cyclegan and yolov3-dense. *Journal of Sensors 2019*.
100. Zeng, M., Gao, H., & Wan, L. (2021). Few-shot grape leaf diseases classification based on generative adversarial network. In *Journal of Physics: Conference Series* 1883(1), 012093, IOP Publishing.
101. Nazki, H., Lee, J., Yoon, S., & Park, D. S. (2019). Image-to-image translation with gan for synthetic data augmentation in plant disease datasets. *Smart Media Journal*, 8(2), 46–57.
102. Chen, Y., Pan, J., & Wu, Q. (2023). Apple leaf disease identification via improved cyclegan and convolutional neural network. *Soft Computing*, pp. 1–14.
103. Li, J., Zhao, X., Zhou, G., Zhang, M., Li, D., & Zhou, Y. (2021). Evaluating the work productivity of assembling reinforcement through the objects detected by deep learning. *Sensors*, 21(16), 5598.
104. Cap, Q. H., Uga, H., Kagiwada, S., & Iyatomi, H. (2020). Leafgan: An effective data augmentation method for practical plant disease diagnosis. *IEEE Transactions on Automation Science and Engineering*, 19(2), 1258–1267.
105. Arsenovic, M., Karanovic, M., Sladojevic, S., Anderla, A., & Stefanovic, D. (2019). Solving current limitations of deep learning based approaches for plant disease detection. *Symmetry*, 11(7), 939.
106. Xu, M., Yoon, S., Fuentes, A., Yang, J., & Park, D. S. (2022). Style-consistent image translation: A novel data augmentation paradigm to improve plant disease recognition. *Frontiers in Plant Science*, 12, 3361.
107. Maqsood, M. H., Mumtaz, R., Haq, I. U., Shafi, U., Zaidi, S. M. H., & Hafeez, M. (2021). Super resolution generative adversarial network (srgans) for wheat stripe rust classification. *Sensors*, 21(23), 7903.
108. Wen, J., Shi, Y., Zhou, X., & Xue, Y. (2020). Crop disease classification on inadequate low-resolution target images. *Sensors*, 20(16), 4601.
109. Katafuchi, R., & Tokunaga, T. (2020). Image-based plant disease diagnosis with unsupervised anomaly detection based on reconstructability of colors. [arXiv:2011.14306](https://arxiv.org/abs/2011.14306).
110. Bhugra, S., Kaushik, V., Gupta, A., Lall, B., & Chaudhury, S. (2023). Anoleaf: Unsupervised leaf disease segmentation via structurally robust generative inpainting. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 6415–6424).
111. Fawakherji, M., Potena, C., Pretto, A., Bloisi, D. D., & Nardi, D. (2021). Multi-spectral image synthesis for crop/weed segmentation in precision farming. *Robotics and Autonomous Systems*, 146, 103861.

112. Espejo-Garcia, B., Mylonas, N., Athanasakos, L., Vali, E., & Fountas, S. (2021). Combining generative adversarial networks and agricultural transfer learning for weeds identification. *Biosystems Engineering*, 204, 79–89.
113. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems* 30.
114. Kerdegari, H., Razaak, M., Argyriou, V., & Remagnino, P. (2019). Semi-supervised gan for classification of multispectral imagery acquired by uavs. [arXiv:1905.10920](https://arxiv.org/abs/1905.10920).
115. Khan, S., Tufail, M., Khan, M. T., Khan, Z. A., Iqbal, J., & Alam, M. (2021). A novel semi-supervised framework for uav based crop/weed classification. *Plos One*, 16(5), e0251008.
116. Slovak, R., Göschl, C., Su, X., Shimotani, K., Shiina, T., & Busch, W. (2014). A scalable open-source pipeline for large-scale root phenotyping of arabidopsis. *The Plant Cell*, 26(6), 2390–2403.
117. Gaggion, N., Ariel, F., Daric, V., Lambert, É., Legendre, S., Roule, T., Camoirano, A., Milone, D., Crespi, M., Blein, T., & Ferrante, E. (2021). ChronoRoot: High-throughput phenotyping by deep segmentation networks reveals novel temporal parameters of plant root system architecture. *GigaScience* 10(7), giab052. [Online]. Available: <https://doi.org/10.1093/gigascience/giab052>.
118. Möller, B., Schreck, B., & Posch, S. (2021). Analysis of arabidopsis root images—studies on cnns and skeleton-based root topology. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1294–1302).
119. Chen, H., Giuffrida, M. V., Doerner, P., & Tsafaris, S. A. (2019). Blind inpainting of large-scale masks of thin structures with adversarial and reinforcement learning. [arXiv:1912.02470](https://arxiv.org/abs/1912.02470).
120. Chen, H., Valerio Giuffrida, M., Doerner, P., & Tsafaris, S. A. (2019). Adversarial large-scale root gap inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*.
121. Yamamoto, K., Togami, T., & Yamaguchi, N. (2017). Super-resolution of plant disease images for the acceleration of image-based phenotyping and vigor diagnosis in agriculture. *Sensors*, 17(11), 2557.
122. Yang, S., Zheng, L., He, P., Wu, T., Sun, S., & Wang, M. (2021). High-throughput soybean seeds phenotyping with convolutional neural networks and transfer learning. *Plant Methods*, 17(1), 50.
123. Scharr, H., Minervini, M., French, A. P., Klukas, C., Kramer, D. M., Liu, X., Luengo, I., Pape, J.-M., Polder, G., Vukadinovic, D., et al. (2016). Leaf segmentation in plant phenotyping: A collation study. *Machine Vision and Applications*, 27, 585–606.
124. Tang, H., Wang, W., Xu, D., Yan, Y., & Sebe, N. (2018). Gesturegan for hand gesture-to-gesture translation in the wild. In *Proceedings of the 26th ACM international conference on Multimedia* (pp. 774–782).
125. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).
126. Talebi, H., & Milanfar, P. (2018). Nima: Neural image assessment. *IEEE Transactions on Image Processing*, 27(8), 3998–4011.
127. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. *Advances in Neural Information Processing Systems* 29.
128. Min, B., Kim, T., Shin, D., & Shin, D. (2023). Data augmentation method for plant leaf disease recognition. *Applied Sciences*, 13(3), 1465.
129. Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research* 911.

Generative Models for Missing Data



Huiming Xie, Fei Xue, and Xiao Wang

Abstract Missing data poses an ubiquitous challenge across a wide range of applications, stemming from a multitude of causes that are both diverse and context-dependent. The prevailing issue is that most advanced data analysis techniques are primarily tailored for complete datasets, thereby underscoring the indispensable need for effective imputation methods. In this chapter, we embark on an extensive exploration of missing data from a statistical perspective, offering a holistic review of its intricate nature. Our investigation encompasses a deep dive into the various mechanisms underlying missing data, shedding light on their ignorability and identifiability—fundamental concepts essential for understanding and addressing this pervasive issue. Moreover, we present a succinct yet comprehensive overview of influential classical imputation methods, showcasing their contributions to the field. Building upon this foundation, we delve into the latest advancements in generative models, a burgeoning area that holds great promise for learning from and imputing missing data. By harnessing the power of generative models, we aim to unlock novel insights and methodologies that can tackle the challenges posed by missing data. Furthermore, we introduce an approach that specifically addresses the critical problem of non-parametric identifiability in nonignorable missing data through the innovative use of generative models. This novel approach aims to overcome the limitations associated with alternative generative models and provides a potential solution to this challenging issue. To enhance the clarity of our proposed method, we supplement our discourse with curated numerical examples that distinguish its effectiveness from other baselines in specific scenarios. Through the exploration, we hope to pave the way for further research and advancements in this critical domain, ultimately leading to more accurate and reliable analyses and interpretations of incomplete datasets.

H. Xie · F. Xue · X. Wang (✉)
Purdue University, West Lafayette, IN, USA
e-mail: wangxiao@purdue.edu

H. Xie
e-mail: xie339@purdue.edu

F. Xue
e-mail: feixue@purdue.edu

Keywords Missing-data mechanism · Missing data imputation · Ignorability · Nonparametric identification

1 Introduction to the Problem of Missing Data

Missing data is a pervasive issue that arises in various applications, and its significance cannot be understated. The causes of missingness can be diverse and context-dependent. For example, in many surveys, some questions are not answered because of privacy concerns, social stigma, or simply because it is time consuming to do so [1, 2]. In the problem of recommender systems, customer ratings may be unavailable as they have never used the products or choose not to provide their opinions[3, 4]. In the medical field, a patient’s record can be missing due to costly information acquisition or health conditions of the patient [5, 6].

The presence of missing data necessitates the adoption of methods that can handle incomplete datasets directly or involve imputing the missing values for subsequent analyses. From a statistical standpoint, one ultimate objective in addressing missing data is to effectively recover the distribution of complete data, enabling robust and reliable analyses that capture the true underlying patterns and relationships in the data. However, most existing data analysis is typically designed for complete data, making imputation methods indispensable for practitioners in the field. This section serves as a comprehensive introduction to the fundamental concepts underlying missing data problems, establishing a solid groundwork for understanding the diverse methodologies employed in missing data imputation.

Patterns of Missing Data As a commonly encountered concept in missing data analysis, missing data patterns refer to the arrangement of observed and missing values within a dataset. Most common patterns found in the literature are univariate, monotone, and nonmonotone missing data [7]. Univariate missing data patterns occur when there is a single variable with missing data. For multivariate missing data, the missing pattern is called monotone if the variables can be arranged in a sequential order, where, once a variable is not observed, all subsequent variables are also missing. This pattern occurs frequently in longitudinal studies due to dropout. The most general case in applications is the nonmonotone missing pattern for multivariate missing data, where there is no way to order the missingness in a monotonic manner, which is the focus of our discussion in this chapter.

1.1 Missing-Data Mechanism

In the statistical literature, consider a p -dimensional space $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_p$. The full data random vector $X = (X_1, \dots, X_p)'$ takes values in \mathcal{X} . Rubin [8] formalized models for missing data mechanisms by introducing a stochastic missing data indicator, which is a binary vector with the same dimensions p as our full data vector X ,

denoted as $R = (R_1, \dots, R_p)'$, with $R \in \{0, 1\}^p$ where 1 corresponds to “observed”, 0 corresponds to “missing”. Such a missing data generating process would yield i.i.d. samples of the pair (R, X) , denoted by $(r^{(1)}, x^{(1)}), \dots, (r^{(n)}, x^{(n)})$ for n samples. We use capital letters to denote random variables or vectors and lowercases for the values of them. A detailed description and summary of the basic notations for missing data used in this chapter can be found in Table 2 in the Appendix.

1.1.1 Two Classes of Models

There are two ways to specify the joint distribution of (R, X) .

Pattern-mixture models Pattern-mixture models [9–11] specify

$$p_{\phi, \pi}(X, R) = p_\phi(X|R)p_\pi(R),$$

where $p_\pi(R)$ is the marginal distribution of missing data pattern, $p_\phi(X|R)$ is the conditional distribution of X given R , and (ϕ, π) are unknown parameters. The term “pattern mixture” means that the marginal distribution of the full data X is a mixture of distributions according to different missing patterns.

Selection models Selection models [8] specify

$$p_{\theta, \psi}(X, R) = p_\theta(X)p_\psi(R|X),$$

where $p_\theta(X)$ is the full data model and the model $p_\psi(R|X)$ corresponds to the **missing-data mechanism**, with (θ, ψ) as unknown parameters. Given the clearer interpretability of missing-data mechanisms and close alignment with notions of ignorability and identifiability, which we will introduce later in Subsects. 1.2 and 1.3, the utilization of selection models is encountered more frequently in the extensive body of literature on missing data [10].

1.1.2 Categories of Missing-Data Mechanism

There are generally three categories of missing-data mechanisms [8].

Missing Completely at Random (MCAR) When R is independent of X , the data is called *missing completely at random* (MCAR). In this case, we can write the joint distribution to be the product of two marginal distributions in both pattern mixture models as $p_{\phi, \pi}(X, R) = p_\phi(X)p_\pi(R)$ and selection models as $p_{\theta, \psi}(X, R) = p_\theta(X)p_\psi(R)$. These two models are equivalent specifications when we assume the same sets of model parameters $\theta = \phi$ and $\psi = \pi$.

Missing at Random (MAR) The data is said to be *missing at random* (MAR) if the missing-data mechanism in the selection models can be written as $p_\psi(R|X) = p_\psi(R|X_{\text{obs}})$, where X_{obs} is a set of completely observed variables in X . The condition for MAR is less restrictive than that for MCAR.

Missing Not at Random (MNAR) When the missing-data mechanism is neither MCAR nor MAR, it is called *missing not at random* (MNAR), which implies that R can potentially depend on the variables in X which are not completely observed.

1.2 Ignorability of Missing-Data Mechanisms

In maximum likelihood inference with selection models, one would be interested in inference on the parameter θ of the full data model using maximum likelihood by maximizing the likelihood of the observed part of the data and the missing indicators. For convenience, we define another indicator $M = 1 - R$ as the opposite of R , implying that $M_j = 1$ represents X_j missing and 0 represents observed. In this way, we can partition a data vector X into the observed part and the missing part, respectively, denoting $X = (X_{(R)}, X_{(M)})$. For the maximum likelihood inference, we integrate out $x_{(m)}$ in the likelihood of complete data x and r :

$$p_{\theta, \psi}(x_{(r)}, r) = \int p_{\theta, \psi}(x, r) dx_{(m)} = \int p_{\theta}(x) p_{\psi}(r|x) dx_{(m)}.$$

Ignorable Missing-data Mechanisms When the missing-data mechanism is MCAR or MAR, by partitioning the selection models, we can move the probability of the missing-data mechanism out of the integration. For MCAR, we have $p_{\psi}(r|x) = p_{\psi}(r)$ so that

$$p_{\theta, \psi}(x_{(r)}, r) = \int p_{\theta}(x) dx_{(m)} p_{\psi}(r) = p_{\theta}(x_{(r)}) p_{\psi}(r).$$

Similarly, for MAR, since $p_{\psi}(r|x) = p_{\psi}(r|x_{\text{obs}})$, with $x_{\text{obs}} \subseteq x_{(r)}$, we have

$$p_{\theta, \psi}(x_{(r)}, r) = \int p_{\theta}(x) dx_{(m)} p_{\psi}(r|x_{\text{obs}}) = p_{\theta}(x_{(r)}) p_{\psi}(r|x_{\text{obs}}).$$

In this way, the likelihood of the observed joint data can be written as a simple product of two likelihoods. When we are interested in learning the parameter θ , we can focus on maximizing the likelihood of the observed data $x_{(r)}$, and completely *ignore* the missing data mechanism. Hence, both MCAR and MAR mechanisms are called *ignorable* [8].

Nonignorable Missing-data Mechanisms When the data are MNAR, the missing data mechanism cannot be separated out from the integration in the likelihood, and thus we should model the joint distribution of X and R together. In real-world applications, MNAR is likely to occur in surveys [12], medical fields [13], recommender systems [14, 15], etc. For example, people experiencing financial difficulties may be more inclined to decline participating in a survey specifically related to their financial incomes [16]. Accordingly, nonignorable missingness refers to MNAR [8].

1.3 Identifiability of Nonignorable Missing Data

When the missing-data mechanism is nonignorable, i.e., data are MNAR, it has been pointed out that without any other assumption the joint distribution of X and R is not identifiable from the observed data alone. It is straightforward to see that there is no sample for $p(X_{(M)}|X_{(R)}, R)$ in any missing data pattern R because they are missing [17]. Therefore, we need to work under the assumption that the full-data distribution falls within a class defined by a set of restrictions:

Robins [18] defined a class of missing data processes called *nonparametric saturated* (NPS), which essentially restricts that there exists 1 – 1 mapping from the set of observed data distributions to the class of full-data distributions.

Definition 1 A class (set) $\{f(r|x)\}$ of missing data processes is **NPS** over a class \mathcal{F} of observed data laws $\{f(r, x_{(r)})\}$ if, for each law $f(r, x_{(r)}) \in \mathcal{F}$, there exists a unique member of the class, say $f^*(r|x)$ and a unique complete data law for X , say $f^*(x)$ such that $f(r, x_{(r)})$ is the marginal distribution of $(R, X_{(R)})$ corresponding to the joint law $f^*(r, x) \equiv f^*(r|x)f^*(x)$. [18]

The property of NPS has also been called a more intuitive term “**nonparametric identifiability**” later in [19–21].

MAR and MCAR have been shown to satisfy this property without further assumptions [22]. However, for MNAR data, as illustrated in the following example, even assuming fully parametric models, identifiability is not guaranteed [23, 24].

Example 1 Consider the following models for a single variable [24]:

Model 1 $X \sim N(1, 1)$, logit $P(R = 1|x) = -3/2 + x$;

Model 2 $X \sim N(2, 1)$, logit $P(R = 1|x) = 3/2 - x$.

In this case, $P(X, R = 1)$ is the same for both models:

$$\phi(x - 1)\sigma(-3/2 + x) = \phi(x - 2)\sigma(3/2 - x),$$

where $\phi(\cdot)$ is the density function of standard Normal distribution, and $\sigma(\cdot)$ is the Logistic function with $\sigma(x) = (1 + e^{-x})^{-1}$. The two models cannot be identified by the observed distribution alone even if we assume the same unit variance Gaussian model for the full data and a logistic missing-data mechanism.

1.3.1 Sufficient Assumptions

Difficult as it may be, there are still ways in which we can find assumptions sufficient for identification. For a single missing outcome, fully observed auxiliary variables, such as shadow variables and instrumental variables, have been used to achieve identification under nonignorable missingness [25–28]. For multivariate missing data, a variety of models have been studied for nonignorable nonmonotone missingness.

Examples include the group permutation missing process [18], which can be represented as a finite sequence of nested MAR processes. Sequential identification of nonignorable missing data [21] designed series of identifying assumptions that allows specifying distinct missing-data mechanisms for various blocks of variables. Discrete choice models [29] were introduced by connecting multinomial missing-data mechanisms with some underlying utility functions. A restriction of the nearest identified pattern was also proposed in the Bayesian inference framework [30]. Graphical models, such as directed acyclic graphs, have been demonstrated to be a valuable tool for describing multivariate missingness and facilitating the study of identification, with numerous identifying conditions discussed in the graphs. [31–34].

Recently, identification for self-censoring models has been studied with completeness conditions in complete cases [1], where self-censoring refers to the missing-data mechanism where the missingness of each outcome depends only on its own underlying value, i.e., $R_j \perp\!\!\!\perp X_{-j} | X_j, R_{-j}$ for $1 \leq j \leq p$. In contrast to the self-censoring model, the no self-censoring model has been considered in several recent works due to its simplicity [2, 35, 36]. The model of no self-censoring assumes that the missingness process of each outcome does not depend on its own value after conditioning on the values of other outcomes and their missingness indicators, formally, $R_j \perp\!\!\!\perp X_j | X_{-j}, R_{-j}$ for $1 \leq j \leq p$. Nonparametric identification has been shown to be achievable in this case under mild conditions.

Albeit the fact that identifiability has been extensively discussed, the proposals of general identifying conditions do not directly lead to practical imputation algorithms in a scalable setting [16], which presents an opportunity for exploration and improvement on current imputation methods. At this juncture, it becomes crucial for us to undertake a comprehensive examination of various imputation methods, commencing with the classical approaches.

2 Classical Imputation Methods

A most brute-force way to bypass missingness in order to obtain a complete dataset is data deletion, including list deletion or case deletion and pairwise deletion [7], where the former assumes MCAR and the latter MAR, without a gateway to MNAR data. However, even if the missing-data mechanisms are assumed correctly to elude bias, deletion would naturally decrease the precision of estimation due to the reduced sample size. Imputation techniques usually come into play to fill this gap.

2.1 *Imputation with Ignorability Assumptions*

Most of the well-studied imputation methods assume that the missing-data mechanism is ignorable, without taking into account the effect of the missing process or

adjusting for the bias it introduces [7, 37–40]. Under the ignorability assumption, some of these methods can achieve good imputation results [7, 41].

An imputation method can be classified as a single or multiple imputation method depending on whether it estimates one or more than one value for each missing entry, respectively [42]. For single imputation, some of the most easily implemented methods include filling out the missing entries with zeros, but there is little justification. Replacing them with the mean or median of a variable also neglects the correlations between the variables. In contrast, simple regression imputation [43] predicts the most likely value of missing data by regressing on other variables, but does not supply uncertainty about the value. Stochastic regression alleviates this problem by means of simple residual variance; yet it still does not capture the joint distribution of all variables. Nonparametric imputation methods were also proposed, trying to integrate the more complex relations among variables. Typical examples include K-nearest neighbors [37], missForest [38] which is an iterative imputation method based on a random forest, and iterative SVD based on matrix completion with iterative low-rank SVD decomposition [39].

Although many single imputation methods have been proposed in the literature, multiple imputation methods are often preferred because they provide an assessment of uncertainty [11, 44]. Multiple imputation consists of producing multiple complete datasets from incomplete data. Then, each completed dataset is analyzed using a complete data method, and the resulting methods are combined to achieve inference. Multiple imputation frameworks often rely on assumptions of statistical models for the data, and drawing samples from them in order to perform the imputation. Early attempts at multiple imputation frameworks relied on simple parametric models such as Bayesian models and mixtures of Gaussians [42, 45]. General approaches include multivariate imputation by chained equations (MICE) [40], also known as fully conditional specification (FCS). It outlines the use of a set of conditional densities for the multivariate imputation model, addressing the incomplete variables on a variable-by-variable basis. Starting from an initial imputation, it draws imputations by iterating over the conditional densities. Joint modeling (JM) [46] is an alternative to FCS when a suitable parametric multivariate distribution can be found such as the multivariate Gaussian or t distribution.

2.2 *Imputation Accounting for Nonignorability*

In recent years, there has been a significant focus on addressing the challenges posed by MNAR data. To handle MNAR data, several approaches have emerged, with a common strategy being the development of joint models that incorporate both observed data and missingness indicators [3, 14, 47, 47–50].

One prominent method used in various domains, such as modern recommender systems for collaborative filtering, is probabilistic matrix factorization [14, 47]. This approach models the data and missingness indicators jointly, allowing for effective imputation of missing values. The use of causal approaches has also gained attention

in handling MNAR data [3, 48, 49]. In these methods, an explicit model of exposure is incorporated, taking a causal perspective on the MNAR mechanism as a confounding bias. Taking causal relationships into account, these approaches aim to mitigate the impact of MNAR on imputation accuracy.

Furthermore, inverse probability weighting methods have been widely employed to address the bias introduced by MNAR in imputation [47, 50]. These methods assign weights to observations based on the probability that they are missing, effectively adjusting for the nonignorability of the missing data. By incorporating inverse probability weights, the imputation process becomes more robust and provides better estimates of missing values.

The combination of these techniques, including probabilistic matrix factorization, causal modeling, and inverse probability weighting, has shown promising results in imputing MNAR data. These advances provide valuable tools for handling missingness in various applications and contribute to a more accurate and reliable analysis of incomplete datasets.

In general, the field of imputation accounting for nonignorability is rapidly evolving, with ongoing research efforts aimed at developing more sophisticated and effective approaches. By addressing the challenges posed by MNAR data, these methods enable researchers to make more informed decisions and draw reliable conclusions from incomplete datasets.

3 Generative Models for Missing Data

In recent times, there has been a notable shift in the approach to data imputation, driven by the emergence of advanced deep generative models [16, 51–58]. These models have attracted the attention of researchers because of their remarkable expressivity and potential to improve the effectiveness of imputation techniques. Using the power of deep generative models, researchers aim to develop more effective and robust methods for handling data imputation scenarios.

Deep generative models, such as variational autoencoders (VAEs) [59] and generative adversarial networks (GANs) [60], offer a rich framework to learn complex data distributions. These models can capture intricate patterns and dependencies in the observed data, making them well suited for imputing missing values in a variety of domains. By training these models on complete data, they can learn the underlying data distribution and generate plausible imputations for missing values. Researchers have explored various strategies to adapt deep generative models specifically for data imputation. For example, VAEs can be augmented with additional components to explicitly model the missing-data mechanism, allowing for more accurate imputations. GANs, on the other hand, can be trained to generate synthetic data that align with the observed data distribution, providing realistic imputations. Furthermore, recent advancements in deep generative models have introduced novel techniques such as normalizing flows and self-attention mechanisms. These techniques enhance the expressivity and modeling capabilities of deep generative models, enabling them

to capture intricate patterns and dependencies in the data more effectively. By incorporating these advances into data imputation methods, researchers aim to improve the quality of imputations and better preserve the underlying structure of the data.

Overall, the exploration of deep generative models for data imputation represents an exciting and rapidly evolving research direction. By harnessing the expressivity of these models and incorporating state-of-the-art techniques, researchers strive to develop more effective, accurate, and reliable imputation methods that can handle missing data in diverse domains.

3.1 Generative Models for Ignorable Missing Data

The majority of existing research on generative models for missing data imputation relies on the assumption of ignorable missingness [51–57]. Within this context, researchers have extensively investigated various generative models, such as GANs, flow-based models, and deep latent variable models (DLVMs). These models have been actively explored as potential solutions for addressing missing data imputation challenges.

3.1.1 GANs

Two popular imputations that are based on modifications to the Generative Adversarial Networks (GANs) architecture are GAIN [52] and MisGAN [53].

GAIN The Generative Adversarial Imputation Network (GAIN) framework [53] employs an adversarially trained imputer optimized to discriminate between fake and real imputations under the assumption of MCAR. The generator (G) is responsible for observing partial components of a real data vector, imputing the missing components based on the observed information, and generating a completed vector. On the other hand, the discriminator (D) aims to distinguish between the observed and imputed components within the completed vector. To guide D in learning the desired distribution, a hint vector is incorporated that provides additional information regarding the missingness of the original sample. This hint enables D to focus on assessing the imputation quality of specific components. Consequently, G learns to generate samples according to the true data distribution.

More formally, for each dimension of the data $j \in \{1, \dots, p\}$, define a new space $\tilde{\mathcal{X}}_j = \mathcal{X}_j \cup \{*\}$ where $*$ is a point not in any \mathcal{X}_j and let $\tilde{\mathcal{X}} = \tilde{\mathcal{X}}_1 \times \dots \times \tilde{\mathcal{X}}_p$. Define a new random vector $\tilde{X} = (\tilde{X}_1, \dots, \tilde{X}_p) \in \tilde{\mathcal{X}}$ according to

$$\tilde{X}_j = \begin{cases} X_j & \text{if } R_j = 1, \\ * & \text{otherwise.} \end{cases} \quad (1)$$

The goal is to generate samples from $p(X|\tilde{X} = \tilde{x})$, where \tilde{x} is a specific realization of \tilde{X} . In this effort, an imputed data vector \bar{X} produced by the generator $G : \tilde{X} \times \{0, 1\}^p \times [0, 1]^p$ is defined as

$$\bar{X} = G(\tilde{X}, R, (1 - R) \odot Z),$$

where \odot denotes element-wise multiplication, and $Z = (Z_1, \dots, Z_p)$ is noise independent of other variables, which is analogous to that in the standard GANs framework except for the dimension here being $\|1 - R\|_1$ to match the target distribution. Naturally, the completed data vector \hat{X} is defined as

$$\hat{X} = R \odot \tilde{X} + (1 - R) \odot \bar{X}. \quad (2)$$

For the discriminator D , instead of classifying the entire generated sample as real or fake in standard GANs, it aims to distinguish the observed and imputed components within a sample so as to recover the missingness indicators R . A hint mechanism, a random variable H taking values in a space \mathcal{H} , is introduced as a necessary condition for the uniqueness of the optimal distribution with respect to D . A value h is drawn according to $H|R = r$ and passed as an additional vector to the discriminator so that we have the function $D : \mathcal{X} \times \mathcal{H} \rightarrow [0, 1]^p$. Here, D is trained to maximize the probability of correctly predicting R , while G is trained to minimize the probability of D predicting R . Therefore, the objective of GAIN is the minimax problem given by

$$\min_G \max_D V(D, G),$$

where the quantity $V(D, G)$ is defined to be

$$V(D, G) = \mathbb{E}_{\hat{X}, R, H} \left[R^T \log D(\hat{X}, H) + (1 - R)^T (1 - \log D(\hat{X}, H)) \right].$$

with \log being element-wise logarithm and dependence on G is through \hat{X} . GAIN has been experimentally shown to outperform other classical imputation methods for imputation under the MCAR setting. However, the stability of training has been noted as one of the concerns [53].

MisGAN Building upon the assumption of MCAR, the MisGAN approach [53] adopts an adversarially trained imputer known as GANs for Missing Data. In this approach, a specific focus is placed on incorporating an explicit model for the missingness mask. By integrating this additional component, MisGAN aims to enhance the imputation process within the MCAR framework.

Specifically, two generator-discriminator pairs (G_r, D_r) and (G_x, D_x) are utilized for the masks and the data, respectively. With MCAR assumed, the two generators have independent noise distributions p_Z and p_W . Two loss functions for the masks and data are defined correspondingly:

$$L_r(D_r, G_r) = \mathbb{E}_{X,R}[D_r(R)] - \mathbb{E}_W[D_r(G_r(W))],$$

$$L_x(D_x, G_x, G_r) = \mathbb{E}_{X,R}[D_x(f_\tau(X, R))] - \mathbb{E}_{Z,W}[D_x(f_\tau(G_x(Z), G_r(W)))],$$

where f_τ is a masking operator filling missing entries with a constant value τ defined to be

$$f_\tau(X, R) = X \odot R + \tau(1 - R).$$

The generators and discriminators are optimized according to the objectives:

$$\min_{G_x} \max_{D_x \in \mathcal{F}_x} L_x(D_x, G_x, G_r),$$

$$\min_{G_r} \max_{D_r \in \mathcal{F}_r} L_r(D_r, G_r) + \alpha L_x(D_x, G_x, G_r),$$

with hyperparameter α . F_x and F_r are defined so that D_x, D_r are both 1-Lipschitz for Wasserstein GANs [61].

The complexity inherent in advanced generative models can offer advantages when dealing with MCAR data during the imputation process. However, leveraging these models for training purposes presents a challenge due to their close association with GANs. It is important to note that GAN-based frameworks often lack the explicit representation of a probability density model for imputation tasks. Consequently, their generalization to MAR and, particularly, MNAR settings may be less pronounced, limiting their effectiveness in these scenarios. Ongoing research efforts are dedicated to addressing these challenges and developing more effective imputation methods that can handle missing data in diverse scenarios.

3.1.2 Normalizing Flow Models

In the realm of missing data imputation, the utilization of normalizing flow models has garnered attention. Among these models, MCFlow [54] stands as an exemplary approach that draws inspiration from Monte Carlo versions of Expectation-Maximization (EM) algorithms within the context of MCAR assumption. Recognizing the challenges posed by the causality dilemma when training models on incomplete data, MCFlow adopts an iterative learning strategy to address this issue effectively.

MCFlow Using the notation in Eq. 1, representing the observed data as $\tilde{x}_{(r)}$, the data imputation task is formulated as a maximum likelihood problem:

$$x^* = \arg \max_x \log p_X(x) \text{ s.t. } x_{(r)} = \tilde{x}_{(r)}. \quad (3)$$

A tractable, explicit density model for $p_X(x)$ is assumed to exist and be in the form of a generative network G effecting a continuous, differentiable and invertible mapping from a sample of interest $x \sim p_X(x)$ lying in space \mathcal{X} to an embedding representation $z \sim p_Z(z)$ in space $\mathcal{Z} \subset \mathbb{R}^p$: $g : \mathcal{X} \rightarrow \mathcal{Z}$ such that $z = g(x)$ and

$$p_X(x) = p_Z(g(x)) \left| \det\left(\frac{\partial g(x)}{\partial x^T}\right) \right|. \quad (4)$$

The model is tractable if both $p_Z(z)$ and the determinant in Eq. 4 above are tractable. Exact sample generation from $p_X(x)$ is feasible by drawing samples $z \sim p_Z(z)$ and setting $x = g^{-1}(z)$. The network G is implemented in the form of normalizing flows. Given complete data, assuming $g(\cdot)$ is parameterized by a set of parameters θ , learning network G is equivalent to obtaining an optimal set of parameters θ^* such that

$$\theta^* = \arg \max_{\theta} \left\{ \log p_Z(g_\theta(\hat{x})) + \log \left(\left| \det\left(\frac{\partial g_\theta(\hat{x})}{\partial \hat{x}^T}\right) \right| \right) \right\}, \quad (5)$$

where \hat{x} is the full data sample calculated as $\hat{x} = r \odot \tilde{x} + (1 - r) \odot \bar{x}$, with \bar{x} being generated (or initialized) data sample.

In addition to the normalizing flow network G trained in an unsupervised manner, the MCFlow architecture incorporates a feedforward neural network H trained in a supervised manner, which maps input embedding vectors \hat{z} to output embedding vectors \bar{z} through function $h_\phi(\cdot)$, where ϕ are the tunable parameters of the network. It is aimed at finding the embedding vector with the largest possible density estimate with observational constraints by minimizing the objective function:

$$\text{MSE}(\hat{x}_{(r)}, \bar{x}_{(r)}) - \lambda \log p_X(\bar{x}), \quad (6)$$

where $\hat{x} = g_\theta^{-1}(\hat{z})$, $\bar{x} = g_\theta^{-1}(\bar{z})$, $\bar{z} = h_\phi(\hat{z})$ and λ is a hyperparameter, which effectively solves the maximum likelihood objective from Eq. 3.

With certain initialization, MCFlow iteratively imputes the missing data using Eq. 3 via learning H , and then updates the parameters of the generative network G based on the generated samples using Eq. 5. This process is similar to the Monte Carlo EM (MCEM) algorithm [62, 63] in that the E step corresponds to the imputation step interpreted as generating samples from the conditional distribution of missing data given observed data and the current parameter estimation $p_X(x|\bar{x}; \theta)$, while the M step is in line with the optimization step that updates the approximation of the posterior of parameters given the observed and imputed values $p(\theta|x, \tilde{x})$. The iterative learning approach employed by MCFlow involves alternating updates between the density estimation and the imputation of missing entries in the training data, which allows the model to gradually learn and capture the underlying data distribution while simultaneously imputing missing values in a principled manner.

Furthermore, MCFlow takes advantage of the flexibility and expressiveness offered by normalizing flows. These models transform a simple distribution, such as a Gaussian distribution, into a more complex distribution that matches the data distribution. By leveraging normalizing flows, MCFlow can capture complex dependencies and patterns in the observed data, leading to more accurate imputations.

The incorporation of Monte Carlo techniques within the MCFlow framework allows for a principled treatment of uncertainty. By utilizing multiple Monte Carlo samples during the training process, MCFlow can estimate the variability and uncer-

tainty associated with the imputed values. This capability provides valuable insights into the reliability of the imputations and allows for more informed decision-making in subsequent data analysis tasks.

In general, MCFlow showcases the potential of normalizing flow models in the context of missing data imputation. While currently focusing on the MCAR assumption, continued advancements in this area hold promise for more accurate and robust imputation methods that can handle missing data challenges.

3.1.3 Deep Latent Variable Models (DLVMs)

Deep Latent Variable Models (DLVMs) enable the approximation of density by optimizing a variational lower bound. Several attempts have been made to extend the variational autoencoder to incomplete data, each focusing on a different aspect [55–57]. It is noteworthy that instead of relying on the MCAR assumption, DLVMs-based imputation methods make the less restrictive assumption of MAR due to its foundation on maximum likelihood inference.

A straightforward way of generalizing DLVMs to incomplete data is to use the evidence lower bound (ELBO) for observed data based on maximum likelihood estimation for ignorable missing data. In DLVMs, a complete-data density parametrized by θ is assumed to be fully factorized given the latent variable Z :

$$p_\theta(x) = \int \prod_{j=1}^p p_\theta(x_j|z)p(z)dz,$$

which implies the conditional independence of the observed and unobserved parts of the data given the latent variable, i.e., $p_\theta(x|z) = p_\theta(x_{(r)}|z)p_\theta(x_{(m)}|z)$, so that with missingness present, the observed data density can be written as

$$p_\theta(x_{(r)}) = \int \int p_\theta(x_{(m)}|z)p_\theta(x_{(r)}|z)p(z)dzx_{(m)} = \int p_\theta(x_{(r)}|z)p(z)dz.$$

Due to the intractability of the integral over the latent variable, a variational distribution $q_\phi(z|x_{(r)})$ is posited as an approximation to the posterior $p_\theta(z|x_{(r)})$ [59], and the evidence lower bound (ELBO) for the log likelihood of the observed data distribution can be derived as

$$\begin{aligned} \mathcal{L}(\theta, \phi) &= \mathbb{E}_{z \sim q_\phi(z|x_{(r)})} [\log p_\theta(x_{(r)}|z)] - KL(q_\phi(z|x_{(r)}) \| p(z)) \\ &\leq \log p(x_{(r)}). \end{aligned} \quad (7)$$

Using neural networks to represent the probabilistic encoders $q_\phi(z|x_{(r)})$ and decoders $p_\theta(x_{(r)}|z)$ respectively, the variational autoencoder (VAE) structure requires same lengths of inputs for all samples, a property which the data with missingness lacks intrinsically.

HIVAE The heterogeneous-incomplete VAE (HIVAE) [55] uses zero imputed missing data as input to the VAE network by recognizing that the output of every neuron in the multilayer perceptron neural network architectures is a nonlinear transformation of a (linear) weighted sum of the inputs, so the output and the derivatives do not depend on the zero entries. It also takes care to account for mixed-type data. Once the model is trained, the imputation of missing data is performed by either sampling from the generative model or using the inferred parameters of the output distribution to impute using the mode of the inferred distribution.

PVAE Alternatively, specifying the inference model $q_\phi(z|x_{(r)})$ without any initialization of the missing entries requires a capacity of handling subsets of different sizes of observed data while sharing parameters. Drawing inspiration from the Point Net (PN) for point cloud classification [64, 65], Partial VAE (PVAE) [56] achieves this through a permutation invariant set function encoding:

$$c(x_{(r)}) := g(h(s_1); h(s_2); \dots; h(s_{(|r|)})),$$

where s_d contains information of the identity e_d and value x_d of the d -th variable for $1 \leq d \leq |r|$, and $|r|$ is the number of observed variables. $h(\cdot)$, represented by a neural network, maps the input s_d to a latent space (different from that of Z); $g(\cdot)$ is the permutation invariant aggregation operation such as max-pooling or summation. e_d is certain predetermined or learnable embedding. One can use different ways to specify s_d , such as concatenation $s_d = [e_d; x_d]$ and element-wise multiplication $s_d = e_d \odot x_d$. The formulation is shown to generalize the VAE for missing data with zero imputation.

MIWAE The missing data importance-weighted autoencoder (MIWAE) [57] is designed to train DLVMs more efficiently when the training set contains MAR data. It builds upon the importance-weighted autoencoder (IWAE) [66] and maximizes a tighter lower bound on the log-likelihood of the observed data:

$$\mathcal{L}_K(\theta, \phi) = \mathbb{E}_{z_1, \dots, z_K \sim q_\phi(z|x_{(r)})} \left[\log \frac{1}{K} \sum_{k=1}^K \frac{p_\theta(x_{(r)}|z_k)p(z_k)}{p_\phi(z_k|x_{(r)})} \right],$$

for $K \geq 1$. When $K = 1$, the bound becomes analytically the same as the VAE objective in Eq. 7. It has been shown that

$$\mathcal{L}_1(\theta, \phi) \leq \dots \leq \mathcal{L}_K(\theta, \phi) \xrightarrow{K \rightarrow \infty} \log p_\theta(x_{(r)}),$$

so that the lower bound approaches the log-likelihood more closely with a larger K .

Monte Carlo techniques are utilized for single and multiple imputation using the DLVMs trained on an incomplete data set. In terms of single imputation, when mean squared error is the performance metric, the optimal imputation will be the conditional mean

$$\mathbb{E}[x_{(m)}|x_{(r)}] = \int \int x_{(m)} p_\theta(x_{(m)}|x_{(r)}, z) p_\theta(z|x_{(r)}) dz dx_{(m)},$$

which can be estimated using self-normalized importance sampling:

$$\mathbb{E}[x_{(m)}|x_{(r)}] \approx \sum_{l=1}^L w_l h(x_{(m),l}),$$

where $(x_{(m),1}, z_1), \dots, (x_{(m),L}, z_L)$ are i.i.d. samples from the proposal distribution $p_\theta(x_{(m)}|x_{(r)}, z) q_\phi(z|x_{(r)})$ which can be sampled by ancestral sampling, and

$$w_l = \frac{r_l}{r_1 + \dots + r_L}, \text{ with } r_l = \frac{p_\theta(x_{(r)}|z_l) p(z_l)}{q_\phi(z_l|x_{(r)})} \quad (8)$$

for $1 \leq l \leq L$. For multiple imputation, using sampling importance resampling [67] with the weights defined in Eq. 8 enables drawing of approximately i.i.d. samples from $p_\theta(x_{(r)}|x_{(m)})$ when L is large. For M imputations, this can be accomplished by first drawing L samples $\{(x_{(m),l}, z_l)\}_{l=1}^L$ ($L \gg M$), and then sampling M imputations with replacement from the set $\{(x_{(m),l})\}_{l=1}^L$ using the importance weights. MIWAE is demonstrated experimentally to be able to produce accurate single imputations and competitive with state-of-the-art classical imputation methods.

DLVMs-based methods for handling missing data offer a compelling probabilistic framework that enables flexible statistical analyses with relatively stable solutions, despite their reliance on approximate inference. Notably, DLVMs overcome the limitations of assuming MCAR, as they relax this assumption and embrace a slightly more realistic MAR mechanism. This sets them apart from other generative models, demonstrating their versatility and adaptability. Moreover, DLVMs exhibit a seamless extension to handle nonignorable missing data by capturing the joint distribution of both the observed data and the missingness indicators. This straightforward extension ensures a comprehensive and holistic approach to address the complexities of missing data scenarios.

3.2 Generative Models for Nonignorable Missing Data

Unsurprisingly, most current works on generative models for nonignorable missing data are built upon DLVMs. One such extension from DLVMs for MAR data is the not-missing-at-random importance-weighted autoencoder (not-MIWAE) [58].

Not-MIWAE Not-MIWAE seeks to incorporate prior knowledge of missingness in the MNAR case such as self-censoring. The method is a generalization of MIWAE employing importance-weighted variational inference and maximizing a lower bound of the joint likelihood of observed data and missingness indicators, as mentioned above. With variational distribution, the log-likelihood of observed data and miss-
ingnes indicators can be written as

$$\begin{aligned} \log p_{\theta, \psi}(x_{(r)}, r) &= \log \int \frac{p_\psi(r|x) p_\theta(x_{(r)}|z) p(z)}{q_\phi(z|x_{(r)})} q_\phi(z|x_{(r)}) p_\theta(x_{(m)}|z) dx_{(m)} dz \\ &= \log \mathbb{E}_{z \sim q_\phi(z|x_{(r)}), x_{(m)} \sim p_\theta(x_{(m)}|z)} \left[\frac{p_\psi(r|x) p_\theta(x_{(r)}|z) p(z)}{q_\phi(z|x_{(r)})} \right]. \end{aligned}$$

Using the idea of importance weighted variational inference [66] yields a lower bound of the log-likelihood as the objective function

$$\mathcal{L}_K(\theta, \psi, \phi) = \mathbb{E} \left[\log \frac{1}{K} \sum_{k=1}^K w_k \right],$$

where for all $1 \leq k \leq K$,

$$w_k = \frac{p_\psi(r|x_{(r)}, x_{(m),k}) p_\theta(x_{(r)}|z_k) p(z_k)}{q_\phi(z_k|x_{(r)})},$$

and $\{z_k, x_{(m),k}\}_{k=1}^K$ are i.i.d. samples from $q_\phi(z|x_{(r)}) p_\theta(x_{(m)}|z)$. A Bernoulli distribution is used for the probability of the missingness indicators given both the observed and missing data:

$$p_\psi(r|x_{(r)}, x_{(m)}) = p_\psi(r|x) = \text{Bern}(r|\pi_\psi(x)) = \prod_{j=1}^p \pi_{\psi,j}(x)^{r_j} (1 - \pi_{\psi,j}(x))^{1-r_j},$$

where $\pi_{\phi,j}(x)$ is the estimated probability of feature j being observed when the full data vector takes on value x .

The imputation uses $\mathbb{E}[X_{(m)}|x_{(r)}, r]$ analogous to that for MIWAE with an additional condition of missingness indicator, which again can be estimated via self-normalized importance sampling. The model performs best experimentally when the missingness model closely matches the true missing-data mechanism, for example, when the mechanism is known to be self-censoring, using logistic regression on each variable with $\pi_{\phi,j}(x) = \sigma(ax_j + b)$ and the sign of a known outperforms alternative specifications.

The direct generalization, crisp interpretation and the compelling results of not-MIWAE under certain nonignorable missing-data mechanism revealed the potential of DLVMs in this area. Nonetheless, the performance under more general or sophisticated MNAR mechanism remains obscure. Besides, the discussion on nonparametric identifiability of nonignorable missing data is left out completely. These can cause practical issues since such knowledge as the missing-data mechanism being self-censoring may not be readily available even if the data is suspected to be MNAR. **GINA** GINA[16] is another deep generative imputation model for MNAR data based on importance weighted autoencoder (IWAE) [66], which additionally takes into account identifiability of the model and incorporates various nonignorable missing-data mechanisms. One difference of the model of GINA from that of not-MIWAE is

that the latent variable is used not only for the data variable, but also for the missing data mechanism. One main claim is that the latent variables could be the cause of both the data and the missingness directly. The model can be written as

$$p_{\theta, \psi}(x_{(r)}, r) = \int \int \prod_{j=1}^p p_{\theta_j}(x_j|z) \prod_{j=1}^p p_{\psi_j}(r_j|x, z) p(z) dz dx_{(m)}.$$

A key assumption that establishes the theoretical results is identifiability of subsets of parameter θ . Furthermore, the reference model $p_\theta(x)$ is assumed to be parametrized by an *identifiable* VAE [16, 68] to satisfy the previous assumption, i.e.,

$$p_\theta(x|u) = \int p_\theta(x|z) p(z|u) dz,$$

where u is some fully observed auxiliary inputs. Using variational inference, an importance weighted lower bound of $\log p_{\theta, \psi}(x_{(r)}, r)$ can be derived as the objective function:

$$\mathcal{L}_K(\theta, \psi, \phi) = \mathbb{E} \left[\log \frac{1}{K} \sum_{k=1}^K w_k \right],$$

where

$$w_k = \frac{p_\psi(r|x_{(r)}, x_{(m),k}, z_k) p_\theta(x_{(r)}, z_k)}{q_\phi(z_k|x_{(r)})},$$

for $1 \leq k \leq K$. The imputation is based on the conditional distribution of missing data given observed data

$$p_\theta(x_{(m)}|x_{(r)}) \approx \int p_\theta(x_{(m)}|z) q_\phi(z|x_{(r)}) dz.$$

Empirical evidence demonstrates that, in specific scenarios of missingness, the performance of the proposed method surpasses that of not-MIWAE, PVAE, and several alternative approaches.

GINA is one of the very first attempts that overcome the limitations of previous methods addressing identifiability in MNAR data (see Subsect. 1.3.1) for scalable imputation by applying a parametric framework, focusing on learning and inference in latent variable models and leveraging the advancements in deep generative models. However, although it established certain identifiability results, a major concern with this approach is that the identification relies on the parametric identifiability assumption of subset of parameters and the existence of fully observed auxiliary variables, which may not be feasible in cases where fully observed variables are unavailable, and when nonparametric identifiability of the missingness model is not guaranteed inherently.

4 Nonparametric Identifiable Generative Models for Nonignorable Missing Data

Generative models for MNAR data face a significant challenge in addressing the nonparametric identifiability issue of the missingness model. Often, these models either overlook this issue or make premature assumptions about the identifiability of model parameters[5, 16, 58]. To address this crucial gap, we propose a novel deep latent variable model that ensures nonparametric identifiability for handling nonignorable missing data. Our approach focuses on both data generation and imputation, aiming to accurately capture the true underlying joint distribution of the complete data. By leveraging deep learning techniques and incorporating deep neural networks, our model enhances its capacity to learn intricate data representations and generate reliable imputations. Moreover, our approach provides a theoretical guarantee for recovering the ground truth joint distribution, overcoming limitations of existing generative models and enabling robust and accurate imputation in MNAR settings.

Inspired by the general no self-censoring model for identification mentioned in Sect. 1.3.1, we assume the ground truth complete data distribution to have a missing-data mechanism that is no self-censoring given a latent variable \tilde{Z} which is independent of all other variables, i.e.,

$$R_j \perp\!\!\!\perp (X_j, R_{-j})|X_{-j}, \tilde{Z}, \quad (9)$$

for $1 \leq j \leq p$. This enables us to establish the identifiability of missing-data mechanism embodied in the following theorem.

Theorem 1 *$p(r|x)$ is nonparametrically identified under Condition (9).*

With a positivity condition that all missing pattern has a nonzero probability of being observed, i.e.,

$$p(R = \mathbf{1}|x) > c, \quad (10)$$

w.p.1 for some constant $c > 0$, where $\mathbf{1}$ denotes the vector of all entries taking the value of 1, it can be shown that the complete data distribution is nonparametrically identifiable leading to an immediate corollary.

Corollary 1 *Under Conditions (9) and (10), the full-data distribution $p(x, r)$ is nonparametrically identified.*

The proofs of Theorem 1 and Corollary 1 can be found in the appendix.

The assumption of no self-censoring is potentially valuable in situations where the data is suspected to be missing not at random (MNAR), but the precise mechanism by which the missingness depends on the observed variables (X) is unknown. Although the no self-censoring assumption restricts the direct dependence of missingness on its own underlying value, it allows for indirect dependency through other variables in the dataset, which merely precludes dependence on one data variable

yet can achieve nonparametric identifiability. Our assumption incorporates latent variables, which differs slightly from the general no self-censoring assumption $R_j \perp\!\!\!\perp X_j | X_{-j}, R_{-j}$ for $1 \leq j \leq p$ (also discussed in Subsect. 1.3.1). In comparison, our Condition (9) disallows conditional dependence between missingness indicators but permits dependence on a latent variable from another source. These two conditions are not mutually exclusive, but both can be reduced to a special case of no self-censoring $R_j \perp\!\!\!\perp X_j, R_{-j} | X_{-j}$ for $1 \leq j \leq p$, where each missingness indicator depends solely on the other data variables and is independent of any other variable given this dependency.

One significant advantage of the assumption of no self-censoring with latent variables is its natural compatibility with deep latent variable models (DLVMs), eliminating the need for approximations such as pseudo-likelihood [35] or other methods to account for mutual dependence among missingness indicators. Instead, we introduce two latent variables, Z and \tilde{Z} , for the data vectors X and the missingness indicators R , respectively. Through importance-weighted variational inference [66] and the learning capability of variational autoencoders (VAEs) [69], it can be shown that this modeling approach possesses desirable theoretical properties for recovering the ground truth joint distribution of the complete data and the missing data indicators (X, R) . In light of these findings, we refer to our model as **LNSC**, which stands for Deep Latent Variable Model for No Self-Censoring.

Together with the latent variable model for the data distribution, our model for the joint distribution of (X, R) is

$$p_{\theta, \psi}(x, r) = \int \int \prod_{j=1}^p p_\theta(x_j | z) \prod_{j=1}^p p_\psi(r_j | x_{-j}, \tilde{z}) p(z) p(\tilde{z}) dz d\tilde{z},$$

where $z \in \mathbb{R}^{\kappa_1}$, $\tilde{z} \in \mathbb{R}^{\kappa_2}$ for some $\kappa_1 \leq p$, $\kappa_2 \leq p$, with priors $p(z) = \mathcal{N}(0, I)$ and $p(\tilde{z}) = \mathcal{N}(0, I)$. After integrating out the missing part, the integrals over missing and latent variables make direct maximum likelihood for observed part of the data intractable. Positing the variational distributions [59] $q_\phi(z|x_{(r)}, r)$ and $q_\lambda(\tilde{z}|x_{(r)}, r)$ as the approximation to posteriors $p_{\theta, \psi}(z|x_{(r)}, r)$ and $p_{\theta, \psi}(\tilde{z}|x_{(r)}, r)$, and using importance weighted variational inference [66] yields the objective function

$$\mathcal{L}_K(\theta, \psi, \phi, \lambda) = \mathbb{E}\left[\log \frac{1}{K} \sum_{k=1}^K w_k\right],$$

where

$$w_k = \frac{p_\theta(x_{(r)}|z_k) p_\psi(r|x_{(r)}, x_{(m),k}, \tilde{z}_k) p(z_k) p(\tilde{z}_k)}{q_\phi(z_k|x_{(r)}, r) q_\lambda(\tilde{z}_k|x_{(r)}, r)},$$

for $1 \leq k \leq K$.

For imputation after training, we follow the practice in [58] to use optimal imputations $\hat{x}_{(m)}$ that minimize the conditional expectation of the loss function as the performance metric given the observed data and the missingness indicators

$\mathbb{E}(L(x_{(m)}, \hat{x}_{(m)})|x_{(r)}, r)$. When employing the squared error as the loss function, the optimal imputation corresponds to the conditional mean $\mathbb{E}(X_{(m)}|x_{(r)}, r)$.

5 Numerical Examples

In our study, we conducted experiments using real-world datasets obtained from the UCI Machine Learning Repository [70]. Specifically, we utilized two datasets: the banknote authentication dataset and the red wine dataset. The banknote authentication dataset comprises 1372 samples, each characterized by 4 continuous variables. These variables represent the variances and entropy of the images found on banknotes, enabling the classification of banknotes as either genuine or counterfeit. Similarly, the red wine dataset consists of 1599 samples, with each sample described by 11 continuous variables. These variables capture various attributes of the tested red wine samples, allowing for the assessment of different characteristics such as acidity levels, sugar content, and other relevant properties. By using these real-world datasets in our experiments, we aimed to evaluate and validate the performance of our proposed methods in practical scenarios. The inclusion of diverse variables and sample sizes provides a comprehensive evaluation framework for assessing the effectiveness and generalizability of our approaches in different data domains.

To introduce missingness into both datasets, we adopted a simulation approach based on the condition of no self-censoring with latent variables. In this simulation process, we generated a set of latent variables $\tilde{Z}_1, \tilde{Z}_2, \tilde{Z}_3 \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$. Subsequently, we incorporated missingness indicators R_j for each variable X_j , where $1 \leq j \leq 3$. The missingness indicators R_j were generated by applying a Bernoulli distribution with a nonlinear transformation. This transformation incorporates the observed data X_{-j} and the generated latent variables $(\tilde{Z}_1, \tilde{Z}_2, \tilde{Z}_3)$ to model the missing-data mechanism. With this simulation setting, all data variables can be missing, and there are no fully observed data variables. By implementing this simulation procedure, we were able to systematically introduce missingness into the datasets while adhering to the no self-censoring assumption and incorporating latent variables.

To examine the impact of different missing rates, we experimented with two relatively high missing rates of 50% and 70%. These missing rates reflect the proportion of missing entries in the dataset. This approach enabled us to evaluate the effectiveness and robustness of our proposed methods in handling missing data scenarios with various missing rates.

Our evaluation metric is the root mean square error (RMSE) of the imputed values compared to the true values. We compare the performance of our method against the other generative models with MNAR assumptions, namely, not-MIWAE [58] and GINA [16]. Two classical imputation methods, including missForest [38] and MICE [40] with ignorable missingness assumptions are also included as baselines. The implementation details are described in the Appendix.

Table 1 Imputation RMSE for UCI data experiments with different missing rates under the missing-data mechanism of no self-censoring with latent variables

Methods	Banknote (50%)	Wine (50%)	Banknote (70%)	Wine (70%)
LNSC	0.680 ± 0.013	0.721 ± 0.008	0.848 ± 0.005	0.857 ± 0.017
GINA	0.729 ± 0.028	0.727 ± 0.036	0.987 ± 0.050	0.939 ± 0.077
not-MIWAE	0.890 ± 0.112	0.952 ± 0.041	1.215 ± 0.049	1.082 ± 0.087
missForest	0.800 ± 0.012	0.900 ± 0.009	1.059 ± 0.007	1.032 ± 0.014
MICE	0.944 ± 0.000	0.938 ± 0.000	0.977 ± 0.000	1.000 ± 0.000

Table 1 displays the numerical results + for mean and standard error of RMSE for different methods, each with 50 runs. In both data experiments involving the no self-censoring missingness scenario, our method consistently demonstrates superior performance compared to the other approaches. This is evident from the evaluation metrics, specifically both the mean and standard deviation of the RMSE. The consistently better performance of our method can be attributed to the limitations of the compared models, which either suffer from unidentifiability or neglect the missing-data mechanism in the context of MNAR data. By outperforming the other models, our method showcases its ability to effectively handle MNAR data by explicitly considering the missing-data mechanism. On the contrary, the limitations of the compared models hinder their ability to accurately capture the complex relationships inherent in the MNAR data, leading to suboptimal imputation results. Our method, on the other hand, addresses these limitations by explicitly modeling the missing-data mechanism with nonparametric identifiability, allowing for more accurate imputations and improved overall performance.

6 Conclusion

Generative models offer a promising and innovative approach to address the challenge of missing data. Throughout this chapter, we have explored the potential of these models for missing data, leveraging their ability to capture complex patterns and generate realistic samples. We have also highlighted their strengths and limitations in handling various missing data scenarios. By employing generative models, researchers and practitioners have the opportunity to overcome certain limitations of traditional imputation methods designed for complete data.

Nevertheless, it is important to acknowledge that generative models for missing data are still an evolving field with ongoing research and challenges to address. Issues such as the stability and generalizability of some of these models require further investigation. Additionally, ensuring the nonparametric identifiability of the models under the nonignorable missingness setting remains an important consideration, for which we have provided one solution and a potential direction.

In addition to advancing the development and exploration of innovative generative architectures to enhance scalability and efficiency in addressing missing data challenges, there exist several compelling research avenues and unanswered questions in the field of generative models for missing data. These potential future directions include, but are not limited to:

- Integration of Alternative Identification Assumptions: Apart from the familiar no self-censoring assumption, incorporating various identification assumptions for nonignorable missing data, as discussed in Subsect. 1.3.1, and leveraging causal inference approaches could further enrich the modeling framework and enable a more comprehensive understanding of the nonignorable nature of missingness.
- Accounting for Uncertainty in Missing Data Modeling: Given the inherent uncertainty surrounding missing data mechanisms, it would be valuable to incorporate approaches that explicitly address this uncertainty within the modeling process. Although multiple imputation partially accounts for uncertainty by assuming correct and unbiased models for the missing data, they do not address the challenge of unknown missing-data mechanisms themselves. Exploring techniques that explicitly capture and incorporate uncertainty in the modeling process could provide more robust and reliable imputation results.
- Enhancing Evaluation Metrics and Benchmarks: Evaluating imputation performance for real missing data can be challenging, as direct assessment is often limited. Current evaluation practices heavily rely on downstream tasks with imputed data, which may not fully capture the true performance. Although some experimental datasets have been introduced, for instance, the Yahoo! R3 dataset [3, 4, 16, 58], there remains a scarcity of resources to comprehensively address this issue. Establishing appropriate evaluation metrics and standardized benchmarks that account for imputation accuracy and uncertainty estimation is critical. Future research can contribute by developing robust evaluation protocols and benchmark datasets that facilitate fair comparisons between different generative models and methods.

In conclusion, generative models presents a powerful toolset for addressing missing data, and their integration into existing methodologies has the potential to enhance the reliability and effectiveness of data analysis. As the field progresses, continued research and refinement of these models will contribute to their wider adoption and the development of practical applications in diverse domains.

Appendix

Summary of Notations See Table 2.

Table 2 Summary of Basic Notations for Missing Data

Notation	Description
A	A random vector
a	The value of the random vector A
$a^{(i)}$	The value of i -th sample of A
A_j	The j -th coordinate of the random vector A
A_{-j}	$(A_1, \dots, A_{j-1}, A_{j+1}, \dots, A_p)$ for a p -dimensional A
$\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_p$	p -dimensional data space
$X \in \mathcal{X}$	Full data random vector
$R \in \{0, 1\}^p$	Missingness indicators, 1(0) means observed (missing)
$M \in \{0, 1\}^p$	Missingness indicators, 0(1) means observed (missing)
X_{obs}	A set of completely observed variables in X
$X_{(R)}$	The observed part of the data vector X
$X_{(M)}$	The missing part of the data vector X
$\tilde{\mathcal{X}} = \tilde{\mathcal{X}}_1 \times \dots \times \tilde{\mathcal{X}}_p$	A data space where $\tilde{\mathcal{X}}_j = \mathcal{X}_j \cup \{*\}$ for $1 \leq j \leq p$
$\tilde{X} \in \tilde{\mathcal{X}}$	$\tilde{X}_j = X_j$ if $R_j = 1$, $*$ otherwise, for $1 \leq j \leq p$

Proof of Theorem 1

Proof Under Condition (9),

$$\begin{aligned} p(r|x) &= \int p(r|x, \tilde{z}) p(\tilde{z}) d\tilde{z} \\ &= \int \prod_{j=1}^p p(r_j|x_{-j}, \tilde{z}) p(\tilde{z}) d\tilde{z} \\ &= \int \prod_{j=1}^p p(r_j|x_{-j}, R_{-j} = \mathbf{1}, \tilde{z}) p(\tilde{z}) d\tilde{z}. \end{aligned}$$

Thus, $p(r|x)$ is a function of the observed data only.

Proof of Corollary 1

Proof Using the odds ratio parametrization [71, 72] with odds ratio function

$$\text{OR}(r, x) \equiv \text{OR}(r, x; r_0 = \mathbf{1}, x_0 = \mathbf{0}) = \frac{p(r|x)}{p(R = \mathbf{1}|x)} \frac{p(R = \mathbf{1}|X = \mathbf{0})}{p(r|X = \mathbf{0})},$$

the full-data distribution [1] can be written as

$$p(x, r) = \frac{\text{OR}(r, x) p(x|R = \mathbf{1}) p(r|X = \mathbf{0})}{\sum_{r'} \mathbb{E}[\text{OR}(r', y)|R = \mathbf{1}] p(r'|X = \mathbf{0})},$$

which is also a function of observed data only.

Implementation Details

In the experiments in Sect. 5, for all three generative models, we set the dimension of the latent space to be $p - 1$ for both datasets. K is set to be 20 for the importance samples. All three models have the same nonlinear structure for the decoder of missingness indicators. We use one hidden layer for both encoder and decoders with dimension 128. Instead of taking a fixed value, the observational noise for the continuous data variables in the decoder is specified as a learnable parameter as proposed in [69]. All methods are trained with Adam optimizer with batch size 16, and learning rate 0.001 for 30000 epochs. During the imputation, the number of importance samples L is set as 10000. For the two classical imputation methods, the `IterativeImputer` from the `sklearn` package in Python is exploited in the experiments, with the default estimator for MICE with maximum iteration as 100 and `RandomForestRegressor` for missForest with the number of estimators as 100.

References

- Li, Y., Miao, W., Shpitser, I., & Tchetgen, E. J. T. (2022). A self-censoring model for multivariate nonignorable nonmonotone missing data. *arXiv preprint arXiv:2207.08535*.
- Malinsky, D., Shpitser, I., & Tchetgen Tchetgen, E. J. (2021). Semiparametric inference for nonmonotone missing-not-at-random data: The no self-censoring model. *Journal of the American Statistical Association*, pp. 1–9.
- Wang, Y., Liang, D., Charlin, L., & Blei, D. M. (2018). The deconfounded recommender: A causal inference approach to recommendation. *arXiv preprint arXiv:1808.06581*.
- Marlin, B. M., & Zemel, R. S. (2009). Collaborative prediction and ranking with non-random missing data. In *Proceedings of the third ACM conference on Recommender systems* (pp. 5–12).
- Ghalebikesabi, S., Cornish, R., Holmes, C., & Kelly, L. (2021). Deep generative missingness pattern-set mixture models. In *International conference on artificial intelligence and statistics* (pp. 3727–3735). PMLR.
- Xue, F., & Qu, A. (2021). Integrating multisource block-wise missing data in model selection. *Journal of the American Statistical Association*, 116(536), 1914–1927.
- Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B., & Tabona, O. (2021). A survey on missing data in machine learning. *Journal of Big Data*, 8(1), 1–37.
- Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3), 581–592.
- Glynn, R. J., Laird, N. M., & Rubin, D. B. (1986). Selection modeling versus mixture modeling with nonignorable nonresponse. In *Drawing inferences from self-selected samples* (pp. 115–142). Springer.
- Little, R. J. (1993). Pattern-mixture models for multivariate incomplete data. *Journal of the American Statistical Association*, 88(421), 125–134.
- Rubin, D. B. (2004). *Multiple imputation for nonresponse in surveys* (vol. 81). Wiley.
- Shrieve, F. M., Stuart, H., Quan, H., & Ghali, W. A. (2006). Dealing with missing data in a multi-question depression scale: a comparison of imputation methods. *BMC Medical Research Methodology*, 6, 1–10.
- Jakobsen, J. C., Gluud, C., Wetterslev, J., & Winkel, P. (2017). When and how should multiple imputation be used for handling missing data in randomised clinical trials-a practical guide with flowcharts. *BMC Medical Research Methodology*, 17(1), 1–10.
- Hernández-Lobato, J. M., Houlsby, N., & Ghahramani, Z. (2014). Probabilistic matrix factorization with non-random missing data. In *International conference on machine learning* (pp. 1512–1520). PMLR.

15. Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). *Recommender systems: An introduction*. Cambridge University Press.
16. Ma, C., & Zhang, C. (2021). Identifiable generative models for missing not at random data imputation. *Advances in Neural Information Processing Systems*, 34, 27645–27658.
17. Rubin, D. B. (1977). Formalizing subjective notions about the effect of nonrespondents in sample surveys. *Journal of the American Statistical Association*, 72(359), 538–543.
18. Robins, J. M. (1997). Non-response models for the analysis of non-monotone non-ignorable missing data. *Statistics in Medicine*, 16(1), 21–37.
19. Vansteelandt, S., Goetghebeur, E., Kenward, M. G., & Molenberghs, G. (2006). Ignorance and uncertainty regions as inferential tools in a sensitivity analysis. *Statistica Sinica* pp. 953–979.
20. Daniels, M. J., & Hogan, J. W. (2008). *Missing data in longitudinal studies: Strategies for Bayesian modeling and sensitivity analysis*. Chapman and Hall/CRC.
21. Sadinle, M., & Reiter, J. P. (2018). Sequential identification of nonignorable missing data mechanisms. *Statistica Sinica*, 28(4), 1741–1759.
22. Gill, R. D., Laan, M. J., & Robins, J. M. (1997). Coarsening at random: Characterizations, conjectures, counter-examples. In *Proceedings of the first seattle symposium in biostatistics* (pp. 255–294). Springer.
23. Wang, S., Shao, J., & Kim, J. K. (2014). An instrumental variable approach for identification and estimation with nonignorable nonresponse. *Statistica Sinica*, pp. 1097–1116.
24. Miao, W., Ding, P., & Geng, Z. (2016). Identifiability of normal and normal mixture models with nonignorable missing data. *Journal of the American Statistical Association*, 111(516), 1673–1683.
25. Miao, W., & Tchetgen, E. J. T. (2016). On varieties of doubly robust estimators under missingness not at random with a shadow variable. *Biometrika*, 103(2), 475–482.
26. d'Haultfoeuille, X. (2010). A new instrumental method for dealing with endogenous selection. *Journal of Econometrics*, 154(1), 1–15.
27. Liu, L., Miao, W., Sun, B., Robins, J., & Tchetgen, E. T. (2020). Identification and inference for marginal average treatment effect on the treated with an instrumental variable. *Statistica Sinica*, 30(3), 1517.
28. Sun, B., Liu, L., Miao, W., Wirth, K., Robins, J., & Tchetgen, E. J. T. (2018). Semiparametric estimation with data missing not at random using an instrumental variable. *Statistica Sinica*, 28(4), 1965.
29. Tchetgen, E. J. T., Wang, L., & Sun, B. (2018). Discrete choice models for nonmonotone nonignorable missing data: Identification and inference. *Statistica Sinica*, 28(4), 2069.
30. Linero, A. R. (2017). Bayesian nonparametric analysis of longitudinal studies in the presence of informative missingness. *Biometrika*, 104(2), 327–341.
31. Fay, R. E. (1986). Causal models for patterns of nonresponse. *Journal of the American Statistical Association*, 81(394), 354–365.
32. Ma, W.-Q., Geng, Z., & Hu, Y.-H. (2003). Identification of graphical models for nonignorable nonresponse of binary outcomes in longitudinal studies. *Journal of multivariate analysis*, 87(1), 24–45.
33. Mohan, K., & Pearl, J. (2021). Graphical models for processing missing data. *Journal of the American Statistical Association*, 116(534), 1023–1037.
34. Nabi, R., Bhattacharya, R., & Shpitser, I. (2020). Full law identification in graphical models of missing data: Completeness results. In *International conference on machine learning* (pp. 7153–7163). PMLR.
35. Shpitser, I. (2016). Consistent estimation of functions of data missing non-monotonically and not at random. *Advances in Neural Information Processing Systems*, 29.
36. Sadinle, M., & Reiter, J. P. (2017). Itemwise conditionally independent nonresponse modelling for incomplete multivariate data. *Biometrika*, 104(1), 207–220.
37. Kim, K.-Y., Kim, B.-J., & Yi, G.-S. (2004). Reuse of imputed data in microarray analysis increases imputation efficiency. *BMC Bioinformatics*, 5(1), 1–9.
38. Stekhoven, D. J., & Bühlmann, P. (2012). Missforest-non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1), 112–118.

39. Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., & Altman, R. B. (2001). Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6), 520–525.
40. Van Buuren, S., & Groothuis-Oudshoorn, K. (2011). Mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45, 1–67.
41. Van Buuren, S. (2018). *Flexible imputation of missing data*. CRC Press.
42. Little, R. J., & Rubin, D. B. (2019). *Statistical analysis with missing data* (Vol. 793). Wiley.
43. Allison, P. D. (2001). *Missing data*. Sage Publications.
44. Rubin, D. B. (1996). Multiple imputation after 18+ years. *Journal of the American statistical Association*, 91(434), 473–489.
45. Audigier, V., Husson, F., & Josse, J. (2016). Multiple imputation for continuous variables using a Bayesian principal component analysis. *Journal of Statistical Computation and Simulation*, 86(11), 2140–2156.
46. Schafer, J. L. (1997). *Analysis of incomplete multivariate data*. CRC Press.
47. Schnabel, T., Swaminathan, A., Singh, A., Chandak, N., & Joachims, T. (2016). Recommendations as treatments: Debiasing learning and evaluation. In *International conference on machine learning* (pp. 1670–1679). PMLR.
48. Wang, Y., & Blei, D. M. (2019). The blessings of multiple causes. *Journal of the American Statistical Association*, 114(528), 1574–1596.
49. Wang, Y., Liang, D., Charlin, L., & Blei, D. M. (2020). Causal inference for recommender systems. In *Fourteenth ACM conference on recommender systems* (pp. 426–431).
50. Wang, X., Zhang, R., Sun, Y., & Qi, J. (2019). Doubly robust joint learning for recommendation on data missing not at random. In *International conference on machine learning* (pp. 6638–6647). PMLR.
51. Wang, Z., Akande, O., Poulos, J., & Li, F. (2021). Are deep learning models superior for missing data imputation in large surveys? Evidence from an empirical comparison. *arXiv preprint arXiv:2103.09316*.
52. Yoon, J., Jordon, J., & Schaar, M. (2018). Gain: Missing data imputation using generative adversarial nets. In *International conference on machine learning* (pp. 5689–5698). PMLR.
53. Li, S. C.-X., Jiang, B., & Marlin, B. (2019). Misgan: Learning from incomplete data with generative adversarial networks. *arXiv preprint arXiv:1902.09599*.
54. Richardson, T. W., Wu, W., Lin, L., Xu, B., & Bernal, E. A. (2020). Mcflow: Monte carlo flow models for data imputation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14205–14214).
55. Nazabal, A., Olmos, P. M., Ghahramani, Z., & Valera, I. (2020). Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, 107, 107501.
56. Ma, C., Tschiatschek, S., Palla, K., Hernández-Lobato, J. M., Nowozin, S., & Zhang, C. (2018). Eddi: Efficient dynamic discovery of high-value information with partial vae. *arXiv preprint arXiv:1809.11142*.
57. Mattei, P.-A., & Frellsen, J. (2019). Miwae: Deep generative modelling and imputation of incomplete data sets. In *International conference on machine learning* (pp. 4413–4423). PMLR.
58. Ipsen, N. B., Mattei, P.-A., & Frellsen, J. (2020). not-miwae: Deep generative modelling with missing not at random data. *arXiv preprint arXiv:2006.12871*.
59. Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
60. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139–144.
61. Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. In *International conference on machine learning* (pp. 214–223). PMLR.
62. Wei, G. C., & Tanner, M. A. (1990). A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*, 85(411), 699–704.

63. Neath, R. C., et al. (2013). On convergence properties of the monte carlo em algorithm. *Advances in modern statistical theory and applications: a Festschrift in Honor of Morris L. Eaton* (pp. 43–62).
64. Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 652–660).
65. Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., & Smola, A. J. (2017). Deep sets. *Advances in Neural Information Processing Systems 30*.
66. Burda, Y., Grosse, R., & Salakhutdinov, R. (2015). Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*.
67. Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian data analysis*. CRC press.
68. Khemakhem, I., Kingma, D., Monti, R., & Hyvarinen, A. (2020). Variational autoencoders and nonlinear ica: A unifying framework. In *International conference on artificial intelligence and statistics* (pp. 2207–2217). PMLR.
69. Dai, B., & Wipf, D. (2019). Diagnosing and enhancing vae models. *arXiv preprint arXiv:1903.05789*.
70. Asuncion, A., & Newman, D. (2007). Uci machine learning repository.
71. Chen, H. Y. (2007). A semiparametric odds ratio model for measuring association. *Biometrics*, 63(2), 413–421.
72. Chen, H. Y. (2010). Compatibility of conditionally specified models. *Statistics and Probability Letters*, 80(7–8), 670–677.

Infrared Image Super-Resolution via GAN



Yongsong Huang and Shinichiro Omachi

Abstract The ability of generative models to accurately fit data distributions has resulted in their widespread adoption and success in fields such as computer vision and natural language processing. In this chapter, we provide a brief overview of the application of generative models in the domain of infrared (IR) image super-resolution, including a discussion of the various challenges and adversarial training methods employed. We propose potential areas for further investigation and advancement in the application of generative models for IR image super-resolution.

Keywords Image super-resolution · Deep learning · Convolutional neural networks (CNN) · Generative adversarial nets

1 Introduction

In modern society, IR images play an irreplaceable role in industry and daily life. Although visible images make it easier to transfer information to people, in specific environments, such as earthquake rescue and security, where there is insufficient light, people have to turn to infrared images. Compared to visible images, infrared images can be better tolerated in tough natural environments and provide rich information about heat sources. Such valuable feedback can help people judge the condition from outdoor equipment and individuals in order to repair damaged infrastructure or help someone in distress. Considering these important applications, high-resolution (HR) IR images are needed urgently. However, IR image resolution is unsatisfactory due to the limitations by current optical devices. With the growing interest in Generative Adversarial Networks (GANs) [12] in the deep learning community, introducing adversarial training methods for IR image super-resolution is on the agenda [14, 26, 38, 54]. In this section, we will present IR imaging applications first, and some

Y. Huang (✉) · S. Omachi

Graduate School of Engineering, Tohoku University, Sendai, Japan
e-mail: huang.yongsong.r1@dc.tohoku.ac.jp

S. Omachi
e-mail: machi@ecei.tohoku.ac.jp

Medical Biochemical Engineering	Vision Tasks	Other Engineering Tasks	
Pharmaceutical industry	Image conversion	Automated vehicle	Food quality control
Medical science	Multispectral matching	Remote sensing	Agriculturas
Cellular observations	Targets detection	Terrain models	Water resource
Fluorescence microscopy	Face recognition	Land surface	Star formation

Fig. 1 IR image super-resolution in different domains [19]

key domain will be discussed. Then the basic components and challenges faced by infrared imaging systems are presented.

1.1 *Infrared Image Applications*

IR image super-resolution is an attractive approach in a wide range of realistic situations. We will present some typical fields of them, such as medical engineering and engineering tasks. Then, other methods will be briefly discussed. IR image super-resolution applications can be seen in Fig. 1 with more details.

In the medical engineering field, degenerative diseases in the nervous system have been gaining great attention. If the mechanisms behind such diseases can be clarified, we will have the opportunity to completely treat diseases that significantly affect the older people's life quality in their later years, such as Alzheimer's disease. Before determining treatment options, understanding the mechanisms involved in these diseases is required for pathological analysis. One of the materials: CRANAD-2, the neurogenic curcumin derivative, is thought to contribute to these studies [6, 50]. IR images can detect CRANAD-2. High-resolution IR images would benefit for NIR nano imaging and further promote the development of correlational studies. Another representative example is the COVID-19 that has received much attention recently. Considering the imbalance in development between different countries and geographies, it will create an unbalanced medical resource. Further, impacting the early detection to COVID-19 in developing countries is that costly lung imaging equipment is not available. Many physicians have attempted to use faster and cheaper X-ray images as a tool for disease diagnosis [8, 37].

IR imaging plays a significant role in the field of engineering. IR cameras can be used to detect temperature changes [21] in systems, which can help identify overheating components that may indicate potential failure or malfunction. In electrical engineering, this can be particularly useful in detecting overheating components that may cause power outages or other issues. IR imaging can also be used to detect leaks

in systems that utilize gases or liquids, such as pipelines [1], allowing engineers to quickly locate and repair these issues and avoid costly damage and downtime. Additionally, IR imaging can be utilized to monitor the performance of systems and equipment, allowing engineers to identify potential problems before they become more serious and take action to prevent costly failures or downtime [7].

Further, IR images have a crucial role in other fields as well. IR imaging can be used to monitor and evaluate the health of ecosystems [36, 43, 59] in the field of environmental protection, particularly in areas where it is difficult to access or study using traditional methods. IR cameras can provide useful information about the health of plants, animals, and other organisms by detecting temperature changes. This enables conservationists and scientists to identify potential difficulties and take measures to protect the environment. And, IR imaging can be used to monitor crops' health and productivity in the agricultural sector [3, 9, 34, 44]. IR cameras can help farmers identify areas of their fields that are not performing as well as others by detecting temperature changes. This enables farmers to take action to increase yields and lower the likelihood of crop failure. Additionally, identify diseases and pests, IR images enabling farmers to safeguard their crops and boost overall productivity. In the food industry, IR cameras can also help identify areas of food that are not being properly cooked or stored by detecting temperature changes. This enables food manufacturers to take action to stop the spread of foodborne illness. As food products are being processed, IR imaging used to monitor their quality [39, 51], allowing manufacturers to spot potential issues and improve product quality. In deep space exploration, IR cameras can provide useful information about the composition and structure of planets, moons, and other objects in the solar system by detecting temperature changes. This helps scientists better comprehend the universe's origins and evolution [5, 40].

Overall, IR imaging is a versatile and powerful tool that can be used in a variety of different fields to provide valuable information and improve our understanding of the world around us.

1.2 *Fundamentals and Challenges in Infrared Imaging*

The link between temperature and the amount of IR radiation emitted by an item is one of the cornerstones of IR imaging [27, 45]. Objects generate more IR radiation at higher temperatures than they do at lower temperatures. Due to this connection, IR cameras may produce pictures based on the amount of IR radiation produced by objects in the field of vision and detect changes in temperature. The requirement for specialist equipment is one of the significant challenges in IR imaging. Special cameras or sensors are needed to find and quantify IR radiation because it is not visible to the naked eye. These cameras and sensors can be expensive, and their efficient use could need specific training. The requirement for meticulous instrument calibration and adjustment is another difficulty in IR imaging. IR cameras and sensors must be properly calibrated to guarantee that they are delivering accurate and dependable data, since they are sensitive to variations in temperature [11, 23, 24, 47, 52]. This

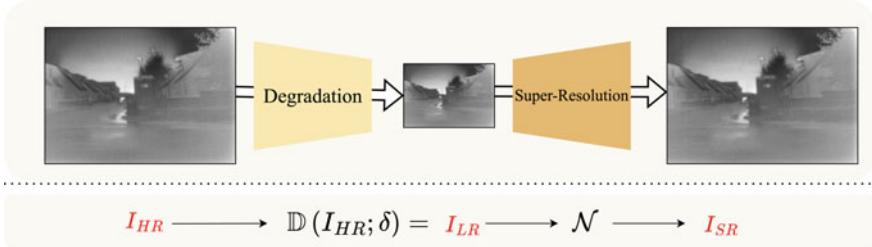


Fig. 2 Degradation and reconstruction in IR image super-resolution. \mathcal{N} denotes neural network

can take a lot of time and requires specialist knowledge. Additionally, a variety of variables, including air conditions, ambient temperature, and the existence of other IR radiation sources, can have an impact on IR imaging. When analyzing and utilizing IR pictures, these aspects must be carefully taken into consideration, since they have the potential to alter the images' accuracy and dependability.

2 Adversarial Training Methods in Super-Resolution

In this section, we will discuss adversarial training methods in super-resolution, including problem definition and adversarial training framework. For problem definition, we will introduce the components in super-resolution. Then, the adversarial training framework in super-resolution will also be shown.

2.1 Problem Definitions

Super-resolution, a technique in the fields of computer vision and image processing, involves the construction of high-resolution images from one or more low-resolution counterparts [10, 32, 42, 55]. This method can be utilized in a variety of contexts, including the enhancement of images from low-resolution cameras or sensors and the augmentation of photographs for medical or scientific study. More details of the problem definition are shown in Eq. 1:

$$I_{LR} = \mathbb{D}(I_{HR}; \delta) \quad (1)$$

where \mathbb{D} denotes a degradation function, I_{HR} is the high-resolution IR image, I_{LR} is the low-resolution IR image and δ is the parameters of the degradation process. Visualization for the IR image super-resolution degradation and reconstruction can be found in Fig. 2.

$$\mathbb{D}(I_{HR}; \delta) = (I_{HR} \otimes k) \downarrow_d; [k, \downarrow_d] \subset \delta, \quad (2)$$

where $I_{HR} \otimes k$ represents the convolution between a blur kernel k and the HR image I_{HR} . In the k , noise and compression are included. And, \downarrow_d is a downsampling factor, e.g., $4\times$ and $8\times$. Briefly, the super-resolution reconstruction objective function of IR images can be described as Eq. 3:

$$\hat{\theta} = \arg \min_{\theta} \mathcal{L}(I_{HR}, I_{SR}) + \lambda \Phi(\theta), \quad (3)$$

where \mathcal{L} denotes the loss function, between the HR image I_{HR} and the SR image I_{SR} . $\Phi(\theta)$ and λ are the regularization term and punishment parameter, respectively. More details about the IR image super-resolution definition can be found in this literature [19].

2.2 Adversarial Training Framework

The issue of super-resolution, which entails raising an image or video's resolution, has been addressed using GANs [12, 14, 26, 38, 54]. In a super-resolution GAN, the discriminative model D is trained to tell the difference between existing high-resolution instances I_{HR} and produced ones I_{SR} , while the generative model G is taught to create a high-resolution version of pictures or videos I_{SR} . A high-quality super-resolution effect is produced by the GAN when generated instances are indistinguishable from actual ones. The two models are trained in tandem using an adversarial process. The generative model ultimately improves greatly at providing realistic examples of high quality thanks to this back and forth training procedure. For super-resolution generative adversarial network (SRGAN), the objective function is shown in Eq. 4. A feed-forward CNN G parametrized by θ_G and discriminator network D which was defined by θ_D , aim to solve the adversarial min-max problem:

$$\begin{aligned} & \min_{\theta_G} \max_{\theta_D} \mathbb{E}_{I_{HR} \sim p_{\text{train}}(I_{HR})} [\log D_{\theta_D}(I_{HR})] \\ & + \mathbb{E}_{I_{LR} \sim p_G(I_{LR})} [\log (1 - D_{\theta_D}(G_{\theta_G}(I_{LR})))] \end{aligned} \quad (4)$$

The visualization of the adversarial generation framework for infrared images is shown in Fig. 3. Typically, we feed the generator G with the low-resolution image I_{LR} , which is the high-resolution sample I_{HR} after downsampling \mathbb{D} . Furthermore, the discriminator D optimizes the objective function to achieve Nash equilibrium with the G and finally generate the super-resolution image I_{SR} .

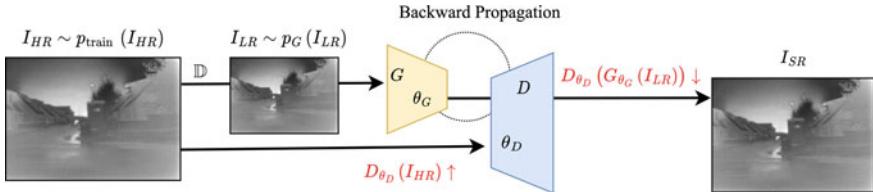


Fig. 3 The adversarial training framework for IR image super-resolution

3 Generative Adversarial Network and IR Image SR

After GAN was proposed, research based on generative models began to emerge in the field of super-resolution. Including SRGAN [26], ESRGAN [54], and various other types of GAN models, they promote the development of this field together through the improvement of the model and the modification of mathematical analysis (WGAN [14]). In the field of IR super-resolution, applications and research based on GAN models are also beginning to appear.

Before this, people have tried to directly use GAN models for normal images in infrared image reconstruction. Researchers directly used SRGAN to reconstruct infrared images and obtained acceptable reconstruction images [48]. However, the blurring of the edges and the lack of clarity of the details are still unacceptable. Uses an improved DCGAN [13] to reconstruct infrared images. This work is different from directly using SRGAN to reconstruct images: first, it targets the reconstruction of face infrared images and proposes a self-built dataset. Secondly, the reconstruction effect is not compared with similar GAN methods, so it is difficult to explain the actual effect. The work of compares the use of SR algorithms on IR images. Including SRGAN, ESRGAN, LapSRN [25], RCAN [63], and SRFBN [31]. The experimental results show that the SRFBN model has the best generalization ability, and for GAN models, there are always unpleasant artifacts due to inherent pattern defects. The GAN model used in normal images with more features has worse edge reconstruction effects in infrared images with fewer patterns. These works all show that the direct application of normal image algorithms in IR image reconstruction may encounter domain shift difficulties.

In the following sections, we describe the generative model in IR image super-resolution methods by module improvement (Sect. 3.1) and introducing extra information (Sect. 3.2), respectively.

3.1 Module Improvement

For module improvement, researchers focus on new modules and novel loss functions. In their work, Rivadeneira et al. [46] propose attention modules and new loss functions. This framework based on Cycle-GAN was proposed and ResNet was

introduced as a module for encoder. The attention module is used after the encoder. They further proposed to use Sobel edge detector as a new loss function to evaluate the similarity between images and to lead the network reconstruction.

Further, more modules related to attention mechanism are studied and proposed. The classical SeNet [16] network is used in IR image super-resolution tasks. Considering the network convergence difficulties, WGAN was introduced which is expected to overcome problems, by the gradient punishment [18, 33]. It is remarkable that the difference between the pixel attention mechanism and the channel attention mechanism will be further shown in the SR image quality.

3.2 *Introducing Extra Information*

After the model improvement, the community started to focus on introducing extra information for the IR image super-resolution task. Because IR images have limited feature information compared to visible images, introducing information from other sources will bridge the gap. For introducing information, the two categories mainly include: hybrid and separated. More details about these two different types of patterns will be discussed below. For the hybrid model, feature information from different sources is not purposefully distinguished. These features are coupled together in the feature space defined by the model parameters.

For example, researchers propose a multimodal visual thermal fusion model that aims to introduce high-frequency information from visible images to help reconstruct thermal images [2]. Experiments have shown that this method using high-frequency information from visible images to reconstruct thermal images is helpful to improve the thermal image quality. There have been improvements in both objective assessment and subjective evaluation.

Moreover, the split model considering that different domain information should be employed more reasonably and the split type approach is further investigated. [17] suggests a strategy using transfer learning to represent the specific patterns in the visible and infrared images in different modules (see Fig. 4). This information is fused in the tail module finally. This strategy illustrates the effectiveness from the view of mathematical analysis for splitting strategies. The experimental results also demonstrate competitive results.

There are some other types of approaches that also help to enhance the image quality. [35] recommends that the image gradient should be more studied with its first layer features map extracted from VGG is used as the backbone as gradient information and further calculated loss function to guide the network to reconstruct the IR image. Other models recommend that the SR task should be considered with references to other tasks, such as denoising and detection [4, 41]. For denoising, the framework proposed by Batchuluun is used for the detection task, but the network layers are deepened in the generator to achieve denoising and super-resolution. The detection framework suggested by [41] is also concerned with the correlation

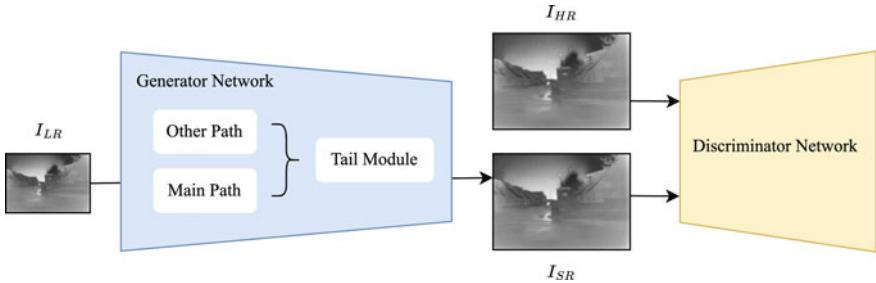


Fig. 4 Transfer learning strategies are used in this framework to fuse the latent space between two domains, the visible image and the infrared image [17]

between multiple tasks. This work also uses a combined loss function to help enhance the image quality.

4 Future Trends

In this section, we will discuss the future trends in IR image super-resolution with GAN. First, new network architectures are helpful for improving the ability to represent patterns. Then, we will present promising directions: image quality assessment. Finally, unsupervised super-resolution in IR images will be discussed.

4.1 Network Design

With the explosive growth of deep learning, more and more new frameworks and models are being proposed and followed. Representative works include the Transformer and diffusion models.

For Transformers, its excellent performance in obtaining long-distance dependency information has made it widely used in various vision tasks and NLP tasks [15, 22, 28, 56, 58, 60]. The results show that the introduction of Transformers into SR tasks has improved both subjective and objective indicators. It should be noted that in the IR image super-resolution task, we focus on the unique patterns of infrared images, such as gradient changes, which will bring new challenges to Transformers. In addition, the data-hungry cloud always hovers over the Transformer. Considering the difficulty of collecting IR image samples, we need to be more careful when introducing difficulties when introducing it into the IR super-resolution field.

On the other hand, the good interpretability of the diffusion model makes it more attractive [17, 18, 54]. More importantly, it is also a generative model. The ability

to fit the data distribution well attracts researchers to use it as a backbone network to build a neural network for IR image super-resolution.

4.2 Unsupervised Image Super-Resolution

Blind super-resolution, which involves reconstructing an HR image from a single LR image, has long been neglected in the field of super-resolution [30, 57, 62, 64]. However, research on blind super-resolution has the potential to address the challenge of real-world super-resolution, where training samples for neural networks are often paired, unlike the degraded LR images that are commonly obtained from imaging devices, particularly in the case of infrared images.

Current approaches to solving the blind super-resolution problem focus on modeling the data distribution as accurately as possible in order to synthesize training data, which is a valuable approach [20, 29, 49, 53, 61]. However, researchers could also consider improving the model by defining the structure of the generator and conducting further mathematical analysis. The interpretability of these types of methods could be beneficial for the continued growth and advancement of the field.

5 Conclusion

In this chapter, we discuss the potential applications and future directions of generative models in the context of IR image super-resolution. The use of generative adversarial networks (GANs) in super-resolution tasks has gained significant attention, particularly in the domain of IR images, which exhibit unique patterns. Researchers have made significant progress in this area through the development of novel module designs and the incorporation of additional information. It is anticipated that new generative models, such as diffusion models, will continue to drive advancements in this field.

References

1. Allred, B., Martinez, L., Fessehazion, M. K., Rouse, G., Koganti, T., Freeland, R., Eash, N., Wishart, D., & Featherlingill, R. (2021). Time of day impact on mapping agricultural subsurface drainage systems with uav thermal infrared imagery. *Agricultural Water Management*, 256, 107071.
2. Almasri, F., & Debeir, O. (2018). Multimodal sensor fusion in single thermal image super-resolution. In *Asian conference on computer vision* (pp. 418–433). Springer.
3. Barzin, R., Kamangir, H., & Bora, G. C. (2021). Comparison of machine learning methods for leaf nitrogen estimation in corn using multispectral uav images. *Transactions of the ASABE*, 64(6), 2089–2101.

4. Batchuluun, G., Kang, J. K., Nguyen, D. T., Pham, T. D., Arsalan, M., & Park, K. R. (2020). Deep learning-based thermal image reconstruction and object detection. *IEEE Access*, 9, 5951–5971.
5. Biller, B. A., Close, L. M., Li, A., Bieging, J. H., Hoffmann, W. F., Hinz, P. M., Miller, D. T., Brusa, G., Lloyd-Hart, M., Wildi, F., Potter, D. E., & Oppenheimer, B. D. (2005). High-resolution mid-infrared imaging of the asymptotic giant branch star rv bootis with the steward observatory adaptive optics system. *The Astrophysical Journal*, 620, 450–458.
6. Bouzin, M., Marini, M., Chirico, G., Granucci, F., Mingozzi, F., Colombo, R., D'Alfonso, L., Sironi, L., & Collini, M. (2022). Melanin concentration maps by label-free super-resolution photo-thermal imaging on melanoma biopsies. *Biomedical optics express*, 13(3), 1173–1187.
7. Butkevich, A. N., Weber, M., Cereceda Delgado, A. R., Ostersehl, L. M., D'Este, E., & Hell, S. W. (2021). Photoactivatable fluorescent dyes with hydrophilic caging groups and their use in multicolor nanoscopy. *Journal of the American Chemical Society*, 143(44), 18388–18393.
8. Canales-Fiscal, M.R., López, R.O., Barzilay, R., Treviño, V., Cardona-Huerta, S., Ramírez-Treviño, L.J., Yala, A., & Tamez-Peña, J.G.: Covid-19 classification using thermal images: Thermal images capability for identifying covid-19 using traditional machine learning classifiers. *Proceedings of the 12th ACM conference on bioinformatics, computational biology, and health informatics*.
9. Cao, Y., Li, G. L., Luo, Y. K., Pan, Q., & Zhang, S. Y. (2020). Monitoring of sugar beet growth indicators using wide-dynamic-range vegetation index (wdvri) derived from uav multispectral images. *Computers and Electronics in Agriculture*, 171, 105331.
10. Chen, H., He, X., Qing, L., Wu, Y., Ren, C., Sheriff, R. E., & Zhu, C. (2022). Real-world single image super-resolution: A brief review. *Information Fusion*, 79, 124–145.
11. Chen, L., Zhou, Z., Xi, N., Yang, R., Song, B., Sun, Z., & Su, C. (2014). Super resolution infrared camera using single carbon nanotube photodetector. In *Sensors* (pp. 1038–1041). IEEE.
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139–144.
13. Guei, A. C., & Akhloifi, M. (2018). Deep learning enhancement of infrared face images using generative adversarial networks. *Applied optics*, 57(18), D98–D107.
14. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of wasserstein gans. *Advances in Neural Information Processing Systems* 30.
15. Herrmann, C., Sargent, K., Jiang, L., Zabih, R., Chang, H., Liu, C., Krishnan, D., & Sun, D. (2022). Pyramid adversarial training improves vit performance. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13419–13429).
16. Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7132–7141).
17. Huang, Y., Jiang, Z., Lan, R., Zhang, S., & Pi, K. (2021). Infrared image super-resolution via transfer learning and psrgan. *IEEE Signal Processing Letters*, 28, 982–986.
18. Huang, Y., Jiang, Z., Wang, Q., Jiang, Q., Pang, G.: Infrared image super-resolution via heterogeneous convolutional wgan. In: Pacific Rim International Conference on Artificial Intelligence, pp. 461–472. Springer (2021)
19. Huang, Y., Miyazaki, T., Liu, X., & Omachi, S. (2022). Infrared image super-resolution: Systematic review, and future trends. <https://doi.org/10.48550/ARXIV.2212.12322>.
20. Huang, Y., Wang, Q., & Omachi, S. (2022). Rethinking degradation: Radiograph super-resolution via aid-srgan. arXiv preprint [arXiv:2208.03008](https://arxiv.org/abs/2208.03008).
21. Jang, K., Jung, H., & An, Y. K. (2022). Automated bridge crack evaluation through deep super resolution network-based hybrid image matching. *Automation in Construction*, 137, 104229.
22. Ke, J., Wang, Q., Wang, Y., Milanfar, P., & Yang, F. (2021). Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 5148–5157).
23. Kim, S. H., Choi, B. S., Lee, J., Lee, J., Park, J. H., Lee, K. I., & Shin, J. K. (2018). Averaging current adjustment technique for reducing pixel resistance variation in a bolometer-type uncooled infrared image sensor. *Journal of Sensor Science and Technology*, 27(6), 357–361.

24. Kong, W., Cao, P., Zhang, X., Cheng, L., Wang, T., Yang, L., & Meng, Q. (2013). Near-infrared super resolution imaging with metallic nanoshell particle chain array. *Plasmonics*, 8(2), 835–842.
25. Lai, W. S., Huang, J. B., Ahuja, N., & Yang, M. H. (2017). Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 624–632).
26. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., & Wang, Z., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4681–4690).
27. Lee, H., Olson, T., Manville, D., & Cloud, G. (2007). Image analysis and understanding using super resolution. In *Display technologies and applications for defense, security, and avionics* (Vol. 6558, pp. 95–101). SPIE.
28. Lee, K., Chang, H., Jiang, L., Zhang, H., Tu, Z., & Liu, C. (2021). Vitgan: Training gans with vision transformers. In *International conference on learning representations*.
29. Lee, S., Ahn, S., & Yoon, K. (2022). Learning multiple probabilistic degradation generators for unsupervised real world image super resolution. arXiv preprint [arXiv:2201.10747](https://arxiv.org/abs/2201.10747).
30. Li, X., Chen, C., Lin, X., Zuo, W., & Zhang, L. (2022). From face to natural image: Learning real degradation for blind image super-resolution. arXiv preprint [arXiv:2210.00752](https://arxiv.org/abs/2210.00752).
31. Li, Z., Yang, J., Liu, Z., Yang, X., Jeon, G., & Wu, W. (2019). Feedback network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3867–3876).
32. Liu, A., Liu, Y., Gu, J., Qiao, Y., & Dong, C. (2022). Blind image super-resolution: A survey and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
33. Liu, Q. M., Jia, R. S., Liu, Y. B., Sun, H. B., Yu, J. Z., & Sun, H. M. (2021). Infrared image super-resolution reconstruction by using generative adversarial network with an attention mechanism. *Applied Intelligence*, 51(4), 2018–2030.
34. Liu, S., Zeng, W., Wu, L., Lei, G., Chen, H., Gaiser, T., & Srivastava, A. K. (2021). Simulating the leaf area index of rice from multispectral images. *Remote Sensing*, 13(18), 3663.
35. Liu, X., Chen, Y., Peng, Z., & Wu, J. (2019). Infrared image super-resolution reconstruction based on quaternion and high-order overlapping group sparse total variation. *Sensors*, 19(23), 5139.
36. Lloyd, D. T., Abela, A., Farrugia, R. A., Galea, A., & Valentino, G. (2021). Optically enhanced super-resolution of sea surface temperature using deep learning. *IEEE Transactions on Geoscience and Remote Sensing*, 60, 1–14.
37. Lukose, J., Chidangil, S., & George, S. D. (2021). Optical technologies for the detection of viruses like covid-19: Progress and prospects. *Biosensors and Bioelectronics*, 178, 113004–113004.
38. Ma, C., Rao, Y., Cheng, Y., Chen, C., Lu, J., & Zhou, J. ((2020)). Structure-preserving super resolution with gradient guidance. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7769–7778).
39. Martínez Gila, D. M., Navarro Soto, J. P., Satorres Martínez, S., Gómez Ortega, J., & Gámez García, J. (2022). The advantage of multispectral images in fruit quality control for extra virgin olive oil production. *Food Analytical Methods*, 15(1), 75–84.
40. Megeath, S. T., Cox, P., Bronfman, L., & Roelfsema, P. R. (1996). Evidence for ongoing star formation in the carina nebula. *Astronomy and Astrophysics*, 305, 296–307.
41. Mostofa, M., Ferdous, S. N., Nasrabadi, N. M. (2020). A joint cross-modal super-resolution approach for vehicle detection in aerial imagery. In *Artificial intelligence and machine learning for multi-domain operations applications II* (Vol. 11413, pp. 184–194). SPIE.
42. Park, S. C., Park, M. K., & Kang, M. G. (2003). Super-resolution image reconstruction: A technical overview. *IEEE Signal Processing Magazine*, 20(3), 21–36.
43. Ping, B., Meng, Y., Xue, C., & Su, F. (2021). Can the structure similarity of training patches affect the sea surface temperature deep learning super-resolution? *Remote Sensing*, 13(18), 3568.

44. Qi, H., Zhu, B., Wu, Z., Liang, Y., Li, J., Wang, L., Chen, T., Lan, Y., & Zhang, L. (2020). Estimation of peanut leaf area index from unmanned aerial vehicle multispectral images. *Sensors*, 20(23), 6732.
45. Rabal, H. J., & Braga Jr, R. A. (2018). Dynamic laser speckle and applications. CRC Press.
46. Rivadeneira, R. E., Sappa, A. D., Vintimilla, B. X., & Hammoud, R. (2022). A novel domain transfer-based approach for unsupervised thermal image super-resolution. *Sensors*, 22(6), 2254.
47. Schutte, K., de Lange, D. J. J., & van den Broek, S. P. (2003). Signal conditioning algorithms for enhanced tactical sensor imagery. In *Infrared imaging systems: Design, analysis, modeling, and testing XIV* (Vol. 5076, pp. 92–100). SPIE.
48. Shao, B., Tang, X., Jin, L., & Li, Z. (2018). Single frame infrared image super-resolution algorithm based on generative adversarial nets. *Journal of Infrared and Millimeter Wave*, 37(4), 427–432.
49. Son, S., Kim, J., Lai, W. S., Yang, M. H., & Lee, K. M. (2021). Toward real-world super-resolution via adaptive downsampling models. *IEEE transactions on pattern analysis and machine intelligence*.
50. Torra, J., Viela, F., Megías, D., Sot, B., & Flors, C. (2022). Versatile near-infrared super-resolution imaging of amyloid fibrils with the fluorogenic probe cranad-2. *Chemistry*.
51. Wang, H., Hu, R., Zhang, M., Zhai, Z., & Zhang, R. (2021). Identification of tomatoes with early decay using visible and near infrared hyperspectral imaging and image-spectrum merging technique. *Journal of Food Process Engineering*, 44(4), e13654.
52. Wang, S. P. (2016). Stripe noise removal for infrared image by minimizing difference between columns. *Infrared Physics and Technology*, 77, 58–64.
53. Wang, X., Xie, L., Dong, C., & Shan, Y. (2021). Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1905–1914).
54. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., & Change Loy, C. (2018). Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*.
55. Wang, Z., Chen, J., & Hoi, S. C. (2020). Deep learning for image super-resolution: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10), 3365–3387.
56. Wei, Y., Hu, H., Xie, Z., Zhang, Z., Cao, Y., Bao, J., Chen, D., & Guo, B. (2022). Contrastive learning rivals masked image modeling in fine-tuning via feature distillation. arXiv preprint [arXiv:2205.14141](https://arxiv.org/abs/2205.14141).
57. Yang, F., Yang, H., Zeng, Y., Fu, J., & Lu, H. (2022). Degradation-guided meta-restoration network for blind super-resolution. arXiv preprint [arXiv:2207.00943](https://arxiv.org/abs/2207.00943).
58. Yang, J., Li, C., & Gao, J. (2022). Focal modulation networks. arXiv preprint [arXiv:2203.11926](https://arxiv.org/abs/2203.11926).
59. Yang, X., Li, Y., Wei, Y., Chen, Z., & Xie, P. (2020). Water body extraction from sentinel-3 image with multiscale spatiotemporal super-resolution mapping. *Water*, 12(9), 2605.
60. Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., & Wu, Y. (2022). Coca: Contrastive captioners are image-text foundation models. arXiv preprint [arXiv:2205.01917](https://arxiv.org/abs/2205.01917).
61. Zhang, K., Liang, J., Van Gool, L., & Timofte, R. (2021). Designing a practical degradation model for deep blind image super-resolution. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 4791–4800).
62. Zhang, W., Shi, G., Liu, Y., Dong, C., & Wu, X. M. (2022). A closer look at blind super-resolution: Degradation models, baselines, and performance upper bounds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 527–536).
63. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., & Fu, Y. (2018). Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 286–301).
64. Zhou, Y., Lin, C., Luo, D., Liu, Y., Tai, Y., Wang, C., & Chen, M. (2022). Joint learning content and degradation aware feature for blind super-resolution. In *Proceedings of the 30th ACM international conference on multimedia* (pp. 2606–2616).

Generative AI for Fire Safety



M. Hamed Mozaffari, Yuchuan Li, and Yoon Ko

Abstract In the field of fire safety, Generative AI presents promising opportunities to enhance prevention, response, and recovery efforts. In this chapter, we explore the potential applications of Generative AI in fire safety. Generative AI benefits the fire safety field in applications including fires simulation and emergency response training, predictive analytics for fire detection and prediction, evacuation planning and optimization, firefighting robotics, and post-fire reconstruction as well as fire investigation. In this chapter, we provided details of two empirical vision-based examples of employing Generative AI for fire safety applications. These examples show how powerful solutions Generative AI models could be for cases where data shortages have been a hurdle for the advancement of AI for fire safety. With all benefits that Generative AI provides us, careful testing, adherence to safety standards, and collaboration between AI experts and fire safety professionals are crucial to ensure the responsible and effective implementation of Generative AI in the context of fire safety.

Keywords Artificial intelligence for fire safety · Deep learning · Flashover detection and prediction · Generative AI in fire safety · Smoke and flame detection · Smart firefighting

Abbreviations

AI	Artificial Intelligence
AR	Auto-Regressive
BM	Boltzmann Machine
DBN	Deep Belief Network
DCNN	Deep Convolutional Neural Networks
GAN	Generative Adversarial Networks

M. H. Mozaffari · Y. Li · Y. Ko (✉)

Fire Safety Unit, Construction Research Centre, National Research Council Canada, Ottawa,

ON K1A 0R6, Canada

e-mail: Yoon.Ko@nrc-cnrc.gc.ca

GAI	Generative AI
GM	Gaussian Mixture
IR	Infrared Radiation
LSTM	Long Short-Term Memory
RGB	Red, Green, Blue
RNN	Recurrent Neural Network
TBM	Transformer-Based Methods
VAE	Variational Autoencoder

1 Introduction

Generative AI is a class of probabilistic Machine Learning (ML) techniques that learn to describe how training data has been generated and generate novel data samples based on the knowledge learned from the training data (i.e., extracted features). Training data samples are also called observations, and each observation consists of many features (e.g., texture, object characteristics, etc. that are called image features). To elaborate the functionality of Generative AI better, assume we have a large dataset of fire truck images. Having access to this training dataset, we can build one generative model that can generate new images of fire trucks that have never existed but still look as real as possible. The ML model is capable of generating novel images given that the model has learned the general rules governing the appearance and characteristics of a fire truck (i.e., image features such as wheel size, red colour, etc.). The flowchart of this example is shown in Fig. 1.

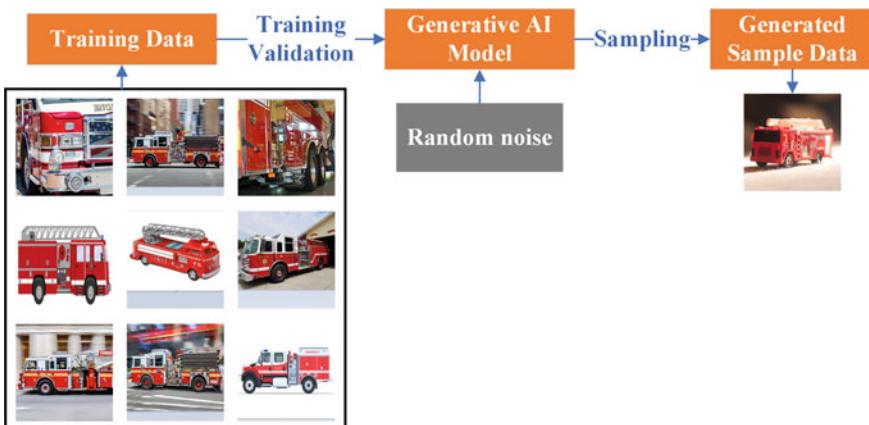


Fig. 1 The process of how generative AI models work in general to generate new distinct images by learning features of training data

In Machine Learning, there is a general consensus that the larger dataset we have, the better the model can extract and generalize image features to a new generated image, whereby generated image could be near to reality and with more details. As such, the first thing that a generative model requires is a training dataset consisting of many samples of the target object tasked to generate. In the case of image analysis and generation, there would be a vast number of ways that the model transforms and combines features learned from training data. Therefore, generating a novel image is a significantly complicated and difficult challenge in computer science. Furthermore, the goal of using a generative model is to create a new image based on best probabilistic analysis of the image features extracted from training data. In other words, the aim is that the model generalizes extracted features in training data for generating one novel and distinct image statistically rather than in a deterministic way. This is the reason it is required to have a random noise in the training of generative models (see Fig. 1).

To generate real and distinct observations, generative models require to have a counterpart called a discriminative model. The discriminative model is trained to predict if a novel-generated observation by the generative model is good enough (i.e., looks like a real image with enough details). Figure 2 shows the general process of training a discriminative model to predict the label of one newly generated image based on prior observations. Generally, discriminative AI models are similar to supervised binary classification machine learning models (e.g., see [1]). Since discriminative model is a supervised technique, training data should be labelled by labels. For example, if the goal is to discriminate between real and fake images, labels can be selected as real or fake (looking at Fig. 2). It is noteworthy to mention that generative models, as we saw, are trained using un-labelled training datasets, and for this reason, they are considered unsupervised methods. See [2, 3] for more details on supervised and unsupervised machine learning methods. Mathematically, we can consider the output of a generative model as probability of observation x , an estimation of the probability that an observation x has been seen before (e.g., features of an image have been learned by the model). Likewise, the output of a discriminative model can be seen as an estimation of the probability that an observation x belongs to category or label y and is modelled by a conditional probability.

Until recently, discriminative AI models have been more readily applicable to engineering problems than generative modelling. The reason is that in engineering, the description of how data was generated is less important than knowing how to categorize observations. For instance, given a wildfire satellite image, fire engineers would care more about the probability that where the pixels of an image contain flames to predict the spread of wildfire. In the medical field, practitioners would want to know the probability that a captured medical image contain anomalies than generating new probabilistic medical images of a patient.

We explained briefly generative and discriminative AI models. These models individually or in combination form the Generative AI field. In general, Generative AI refers to a form of Artificial Intelligence designed to produce a diverse range of data, including images, videos, audio, text, and 3D models. It achieves this by assimilating patterns from existing data and leveraging this knowledge to generate

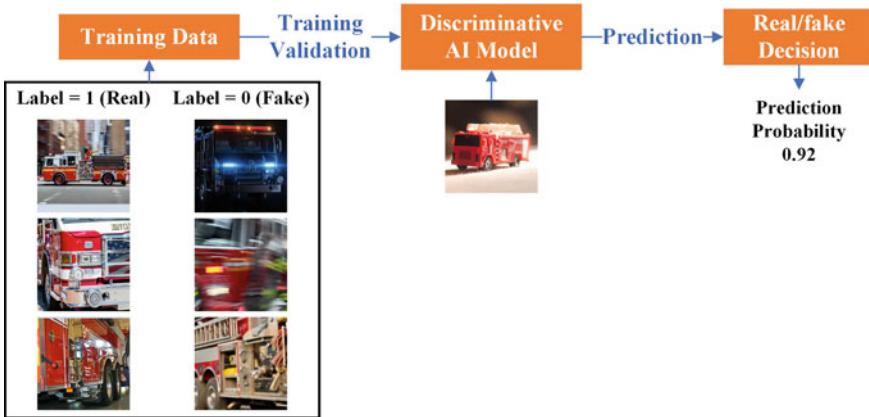


Fig. 2 The process of how discriminative models work in general to distinguish real from fake observations

fresh and distinct outcomes. Generative AI exhibits the ability to create intricate and remarkably realistic content that imitates human creativity, rendering it a valuable resource across various industries such as gaming, entertainment, and product design. Notably, recent advancements in the AI field have significantly enhanced the capabilities of Generative AI models, such as generating new text, image, and video data automatically without human intervention. These breakthroughs have unlocked new opportunities for utilizing Generative AI to tackle intricate problems, foster artistic expression, and even contribute to scientific investigations. In this chapter, we are highlighting benefits of Generative AI models for the problem of fire safety in construction. In the following sections of this chapter, Sect. 2 reviews briefly common Generative AI methods and applications. Examples of employing Generative AI methods in fire safety literature are explained in this section. Section 3 focus on explanation of two popular Generative AI models, Generative Adversarial Networks (GANs) and Long-Short Term Memory (LSTM) applied for detection and prediction of rapid-fire growth in room fire incidents. Section 4 will conclude this chapter and provides future path of Generative AI methods to solve fire safety problems and challenges.

2 Generative AI for Fire Safety

2.1 Categories of Generative AI Methods

Explaining details of all Generative AI methods is out of scope of this chapter. There are plenty of methods in Generative AI field. Curious readers can find popular methods in summary in [4] or in details in [5]. Table 1 enumerate popular methods

Table 1 Name of popular generative AI models

Generative AI method	Sample references
Hidden Markov model	[6]
Naïve Bayesian model	[7]
Gaussian mixture (GM) model	[8]
Boltzmann machine (BM)	[9]
Deep belief network (DBN)	[10]
Generative adversarial network (GAN)	[11]
Auto-encoder (AE)	[12]
Auto-regressive (AR)	[13]
Recurrent neural network (RNN)	[14]
Variational auto-encoder (VAE)	[15]
Transformer based methods (TBM)	[16]

in this field. It is noteworthy to mention that each method has several variants, and sometimes the first original work is not clear. Also, there is no solid line between Generative AI models and discriminative models. Sometimes, they can be used interchangeably. For instance, Recurrent Neural Networks (RNNs) are considered more as discriminative models since they are trained on labeled data. However, since the output of these models are new generated data, they are considered as Generative AI models too. In this chapter, we will explain more details for two variants of Generative Adversarial Network (GAN) and RNN. These constitute the foundational building blocks for many sophisticated Generative AI models currently employed in both industrial and academic settings.

Generative AI holds promise for diverse sectors, including art, writing, software development, healthcare, finance, gaming, marketing, and fashion. The early 2020s witnessed a substantial increase in investment in Generative AI, with major corporations like Microsoft, Google, Meta, and Nvidia, along with numerous smaller firms, actively working on developing Generative AI models [17]. Application-based, we categorized Generative AI methods into three main prominent categorizes: Generative Adversarial Networks (GANs) [18], Transformer-Based Models (TBMs) [19], and Variational Autoencoders (VAEs) [15].

2.1.1 Generative Adversarial Networks (GANs)

GANs are those generative models that uses deep learning methods to generate data. Deep learning is a branch of machine learning where neural networks have deeper/larger network structure. This deeper structure enables the model to extract better abstract features from the data. Main mechanism of GANs is a competition between two networks, one to generate new data and another one to determine the quality of the generated data. The details of a GAN model is explained in the next section. GAN models are more popular in image processing applications, such as image-to-image

translation [20] like converting or transforming images of real fire trucks into images of toy fire truck.

A cornerstone of the success achieved by Generative Adversarial Networks (GANs) lies in their adversarial structure involving two distinct models [11]. This clever approach transforms the task of training a generative model into a supervised learning problem with two sub-models: the generator and the discriminator. The generator model, which is trained to generate novel instances, and the discriminator model, which aims to distinguish between real examples from the given domain and generated ones, comprise these two sub-models. These models are collaboratively trained in a zero-sum game, or an adversarial setting, until the discriminator model is deceived approximately half of the time. This indicates that the generator model is adept at producing plausible examples, underscoring the strength and efficacy of this adversarial mechanism in GANs. Therefore, GANs can generate fake images near to reality such that even human can't distinguish between real and fake images. Endless application can be imagined using GANs when the goal is to generate new images from already available image datasets.

2.1.2 Transformer Based Methods (TBMs)

TBMs are neural networks that work by learning context and meaning by closely tracking relationships in sequential data, especially when the context matters. TBMs came from natural language processing field, and they are more powerful in text analysis and generation. TBMs are often used to translate or generate texts since texts are more than just words chunked together [21].

The self-attention layer serves as a crucial component in the architecture of a TBM, playing a pivotal role in its computation and final outcomes [22]. By leveraging a technique commonly referred to as self-attention, transformers can discern distant data relationships and mitigate the vanishing gradient problem. This mechanism allows a TBM to identify and analyze relationships between two words, regardless of their distance within a particular context. For instance, in Natural Language Processing (NLP) tasks, the self-attention process quantifies the relevance of a particular word in correlation to its neighboring words within a sentence. Hence, it bolsters the model's ability to understand complex dependencies and relations, augmenting its overall performance.

Popular methods in this category are GPT models using cognitive attention to understand the language context or images and generate new texts and images from massive datasets. For instance, TBMs can translate text and speech in near real-time, describing one image as a paragraph or even writing new contexts composing new music or generating images from text. Overall, when the goal in an application is to generate new data (usually in future time) from existing sequential or temporal data, where a context-aware manner is necessary, selecting TBMs would be a better selection than GAN.

2.1.3 Variational Auto-Encoders (VAEs)

VAEs are the third category of Generative AI methods, and also similar to GANs. GANs are made of two neural networks (i.e., generative and discriminative) competing to generate real new images whereas VAEs have two neural networks (i.e., encoder and decoder) [23].

Contrary to GANs, the encoder and decoder components in VAEs serve a purpose akin to traditional Autoencoders (AEs): the reduction of dimensionality [24]. Dimensionality reduction refers to the process of decreasing the number of features used to represent data. This reduction is achieved through either selection, where only a subset of existing features is retained, or extracted, wherein a smaller number of new features are derived from the existing ones. This process proves beneficial in various scenarios that necessitate low-dimensional data, including data visualization, storage, and computations that are computationally intensive. Thus, VAEs offer a different approach to data processing and representation compared to GANs, underlining the diversity in machine learning methodologies. As a consequence, the aim of VAEs is to regenerate input data in the output side, and find a feature map between the output and input. In other words, encoder extracts features from input data (the output features also called latent space), and decoder reconstructs the input data from the extracted features by encoder.

Moreover, the encoder in a VAE aims to compress the input features into a data distribution defined by its mean and variance, a process that differs from traditional AEs, which map input to a specific data point [25]. Consequently, the decoder reconstructs the output based on a sampled result from the data distribution produced by the encoder. VAEs' inherent characteristic as a probabilistic model facilitates the generation of output that, while different, remains similar to the original input. This distinctive trait highlights the strength and potential of VAEs in the realm of generative modeling, allowing them to create variations of input data while maintaining its core characteristics. VAEs are used in many applications, such as down sampling of data, denoising, dimensionality reduction, and many more. VAEs can also generate new data samples by inserting noise in latent space.

2.2 Application of Generative AI in Fire Safety

2.2.1 Existing Applications of GANs for Fire Safety

The use of Artificial Intelligence and Computer Vision for fire and smoke detection has been a recent research focus [26]. We spotlight several recent works in this field that have applied Generative AI to Fire Safety, with a particular emphasis on image and sensor data. The Generative Adversarial Network (GAN) is one of the most frequently employed Generative AI models in Fire Safety Research. GAN models are capable of creating high-resolution, realistic synthetic images, which is an attribute of crucial importance in Fire Safety Research. Consequently, the way GAN

models work, leads to two primary application trajectories: image generation (for data augmentation) and image translation (for transformation). These applications exemplify the impactful integration of Generative AI in Fire Safety Research.

GANs for Image Generation and Augmentation in Fire Safety

Numerous studies have already employed GAN models for image generation and augmentation. Dung et al. proposed a model for high-quality fire scene image generation [27]. Their model comprises two components: a Fire Kernel Generating Network and a Blending Network. The Fire Kernel Generating Network, based on the StarGAN model, is tasked with fire kernel generation, guided by supervision information as a style reference. The Blending Network is designed to transplant the fire kernel into a scene, fine-tuning the texture and style of the blending region to create a realistic image. Moreover, they embedded a self-attention module into the Blending Network to enhance edge gradient. Qin et al. introduced an FGL-GAN, based on GAN models, for high-quality flame image composition [28]. Their FGL-GAN employs a hierarchical Global–Local generator structure, which locally renders a high-quality flame halo and reflection while maintaining a globally consistent style. The model also incorporates a fire mask as part of the input for the generation module, improving the rendering quality of flame halo and reflection. FGL-GAN adopts the concept of contrastive learning for loss function design to expedite network fitting and reduce blurriness.

GANs for Image Translation and Conversion in Fire Safety

GAN models are also frequently used in Fire Safety Research for image translation, converting images between two distinct domains. Yun et al. built a flashover prediction model which, at its core, utilizes image-to-image translation by a CycleGAN from the vision domain to the IR domain [29]. Trained on pairs of vision and IR images, the model can transform RGB vision images of fire into corresponding IR images. The temperature information extracted from IR images is vital for flashover estimation during fire development. Kacker proposed a model for fusing fire images from multiple domains [30]. Their model can blend RGB vision images and IR images while preserving both visible light and temperature information. Consequently, the temperature information acts as supplementary data, enabling a more robust analysis of fire scenes. Furthermore, GAN models serve as data augmentation techniques, enhancing existing datasets for downstream AI tasks such as smoke and flame detection. Park et al. presented a flame detection framework based on a GAN model for data augmentation [31]. Their GAN model can blend wildfire images into various shapes by inserting damage into a free-wildfire image. They also utilized a weakly supervised object localization mechanism with a gradient-weighted activation map for improved robustness and efficiency. Cheng et al. proposed a framework

for smoke detection [32]. Their model uses a GAN for future smoke frame generation. It accepts a sequence of smoke images as input and generates one image for the future. The GAN model is applied several times in their framework to create a heatmap of future smoke region variation, which aids in smoke detection even with blurry or low-resolution input images.

2.2.2 Existing Reviews on Generative AI Models for Fire Safety

Several review papers on AI models for Fire Safety Research have been published in recent years. Geetha et al. conducted an extensive review of fire detection approaches and applications, encompassing both feature engineering and AI-based methodologies developed in recent years [33]. For Generative AI models, they offer a review of RNN models used in fire detection applications. They discovered that RNN models are proficient at extracting more robust smoke features across multiple domains in sequential frames, significantly enhancing accuracy while reducing false detection rates. However, these improvements are accompanied by a substantial increase in network size compared to other models like CNNs. This may necessitate considerable computational power for processing. In Chaturvedi et al.'s survey of smoke detection techniques [34], GAN models are specifically cited as a prevalent data augmentation approach for downstream tasks, such as smoke detection in Fire Safety Research. GAN models have proven effective in generating datasets featuring various objects in high-quality output images. This is particularly beneficial for obtaining images of smoke and flames, which are typically challenging to capture.

However, thermal images are captured and generated by specialized IR cameras, which exhibit a more intricate imaging principle. This complexity may necessitate training for non-expert users to comprehend the various settings required for different applications, not to mention the potential hardware limitations posed by the sensor. Consequently, Generative AI models like GANs have emerged as an alternative approach for generating IR thermal images in situations where thermal IR cameras may not be readily available.

3 Two Generative AI Examples in Fire Safety

There are plenty of examples for application of AI in Fire Safety, such as [26, 33]. Here, we highlighted two recent instances of using Generative AI in fire engineering using vision data. In the first example, GAN is used for translating RGB images, captured from room fire incidents by a regular camera, into thermal IR images without using any IR camera. The second example shows the power of Generative AI in prediction by generating images few seconds in future for detection and analysis of rapid fire growth phenomena few seconds in advance.

3.1 *GAN for Generation of Infrared (IR) Images from RGB Images*

Generative AI models, such as GANs [18], possess the ability to learn complex patterns and information from existing datasets that traditional mathematical models, such as regression models [35], may struggle to handle. As explained in Sect. 1, these generative models can subsequently use the learned information from training data to guide the generation process.

In the field of fire safety research, thermal information contained within infrared (IR) thermal images serves as a valuable data source that not only supplements RGB visual images of a fire but also surpasses the capabilities of traditional models [36]. Infrared (IR) thermal frame generation from RGB data is an image translation task for GAN models, where the source domain consists of RGB images and the target domain comprises IR thermal images [37–39]. Unlike typical image translation tasks, such as image super-resolution [40] and style transfer [41], images from the source and target domains are not aligned at the pixel level due to differences in sensors and lenses between RGB and IR thermal cameras. Consequently, generation cannot be performed in a pixel-to-pixel manner, where the information of each pixel directly contributes to a single pixel in the generated images.

To address this challenge, a cycle structure borrowed from CycleGAN [42] is employed, enabling the translation of RGB images back to their original form to maintain pixel-level alignment. Additionally, foreground and background attention mechanisms [16] are introduced to better identify flame and smoke patterns, distinguishing them from other illumination sources and dark backgrounds. The overall architecture of this Dual Attention GAN (DAGAN) [43] is depicted in Fig. 3. In this example, DAGAN has been used for generating IR images from RGB images, applicable for detection and even prediction of rapid-fire growth (e.g., flashover [38]) in room fire situations. This network structure is similar to one standard GAN [18] made up of generative and discriminative models. In DAGAN model, generative part receives training RGB images, extract features benefiting from attention mechanism, and generate corresponding IR image. Discriminative part of the DAGAN then reproduces original RGB image from the generated IR image. In other words, DAGAN acts as a VAE but instead of encoder-decoder, it has generative and discriminative sub-models.

In this example [43], the DAGAN model was trained and tested on a dataset comprising 1800 image pairs of RGB and IR data, featuring flame and fire images captured during 17 fire experiments [44]. These experiments included one open fire test, 14 single-item burning tests, and two room fire tests. In the open fire test, a metal box filled with wood studs was ignited, and the flame development was recorded using both a vision camera and a thermal IR camera.

Few representative image pairs from the dataset are displayed in Fig. 4. The RGB and IR image pairs used in training are not pixel-aligned but are synchronized, signifying that each pair was captured from different viewing angles but at the same time. IR images employ a 1024-level constant colorbar ranging from 280

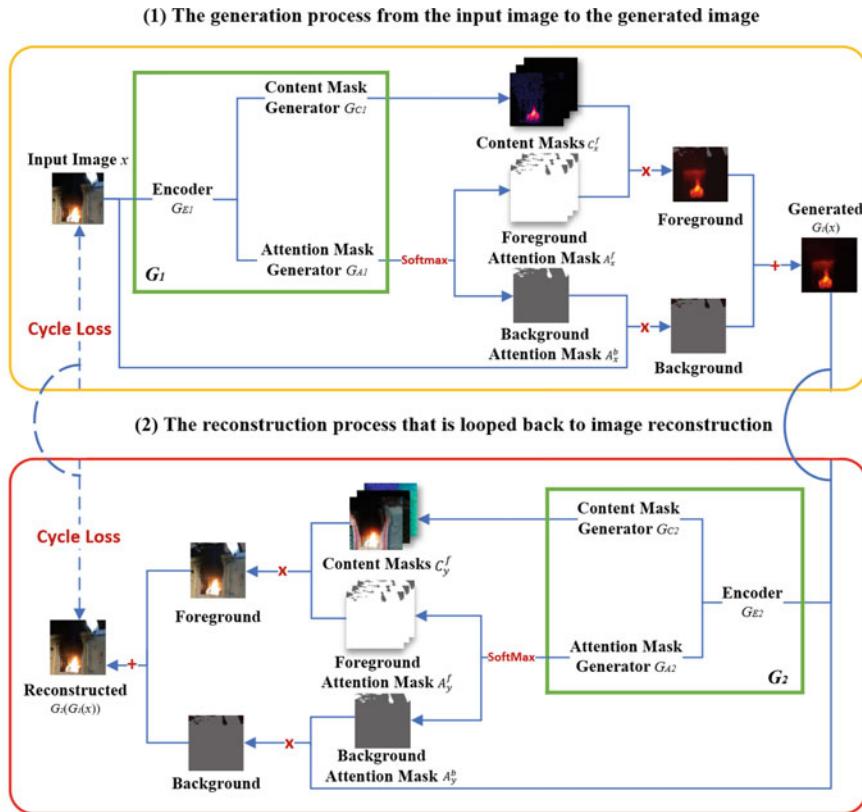


Fig. 3 Architecture of DAGAN for translating RGB images into IR images [45]

to $1400\text{ }^\circ\text{K}$ to convert temperature information into the color domain. This colorbar is depicted in Fig. 4. Results of DAGAN model revealed that it is possible to convert RGB images into IR images even when view angle of images is totally different. Figure 5 presents a selection of input RGB flame images from four fire experiments, alongside a comparison of the translated thermal images generated by DAGAN [43], CycleGAN [42], and AGGAN [46]. The Ground Truth (GT) images, which represent the actual IR data recorded during the fire experiments, are also included for reference. As illustrated in Fig. 5, the quality of thermal images generated by DAGAN is consistently superior to those produced by CycleGAN [42], exhibiting a significantly higher resemblance to the GT images in both foreground (e.g., flame area) and background regions. CycleGAN's outputs display only limited color conversion and lack discernible feature linkages between the input RGB images and the corresponding output thermal images.

While DAGAN demonstrates consistent performance, AGGAN's performance varies depending on the fire development stage. For instance, during the early stages of a fire, as seen in the second row of Fig. 5, AGGAN and DAGAN exhibit similar

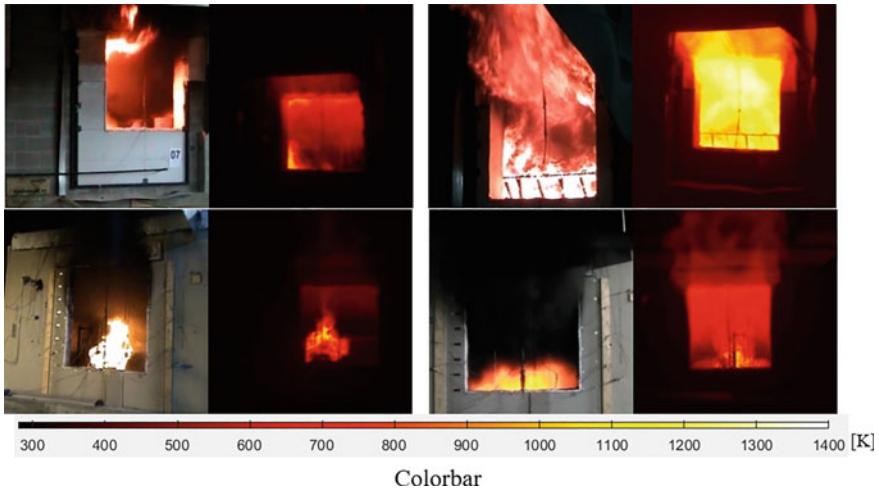


Fig. 4 Samples of image cuts used for DAGAN. The image pairs are from (left to right, up to down) test cases PRF-07, PRF-12, 09-SI-25, and 08-SI-04 [44]

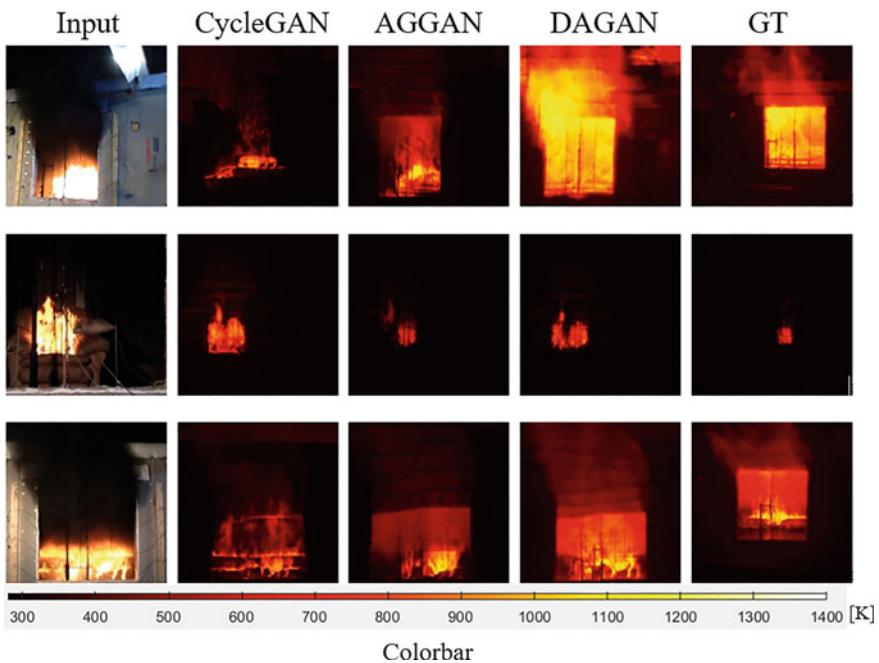


Fig. 5 Samples of images by CycleGAN (second column from left), AGGAN (third column from left) and DAGAN (fourth column from left) for selected cases 31-SI-13 (first row), 08-SI-04 (second row) and 14-SI-06 (third row) [44]. GT is the ground truth of IR images (fourth column)

conversion quality for the background, though DAGAN captures more details of the flame pattern. As the fire intensifies and a smoke layer becomes visible to the human eye, as in the RGB input images in the first and third rows, DAGAN outperforms AGGAN in translating thermal information in the darkened areas caused by the smoke layer. This superiority is evident in the outputs displayed in Fig. 5. Furthermore, DAGAN consistently surpasses AGGAN in generating thermal images with higher accuracy, particularly in the luminous flame areas, as observed in fully developed fire cases like those in the first row of Fig. 5. Additionally, as detailed in the subsequent section, DAGAN-generated images exhibit better temperature accuracy, making them more compatible with the corresponding Ground Truth (GT) images.

Detailed qualitative evaluation of DAGAN comparing with other GAN model, such as CycleGAN and AGGAN can be found in the original study [43]. Another important conclusion from this example is that attention mechanism boosts different Generative AI models, and GAN is not an exception. Our anticipation is that in future using attention mechanism, pattern recognition problems will be solved, and Generative AI models such as GAN will be seen more in challenging engineering and non-engineering applications.

3.2 *Rapid Fire Future Frame Generation Using RNN*

In this fire safety example, another Generative AI model, the powerful Convolutional Long Short Term Memory (ConvLSTM) architecture [47], one variant of RNN networks, generates future images of rapid-fire growth from observation of past events. For a better understanding of ConvLSTM in this example, RNN and LSTM should be explained first in detail. As we discussed in Sect. 2, TBM_s are advanced versions of RNNs with the attention mechanism, yet new to the field of fire safety. This example shows the benefits of RNN models for the detecting and predicting hazardous phenomena in vision data of fire incidents. RNNs [48] are amongst the first artificial neural network structures designed for handling sequential data. These network models show acceptable performance in short-term future predictions. RNNs are similar to feedforward neural networks, except they have a cyclic connection which enables the model to possess the capacity to update its current state based on the observation of past states and current input data [48]. Like other neural networks, RNNs consist of neurons/cells, simplest possible of which are exemplified in Fig. 6. As illustrated, the output of a recurrent layer (see Fig. 6a) is fed to the input in a loop. In reality, this small recurrent network with one layer can be represented in an unrolled version, as shown in Fig. 6b. The main components in a recurrent cell are shown in Fig. 6c. In this figure, x , y , b , and h are input, output, bias, and hypothesis of the model in different time steps. Detailed description of RNN models are not in the scope of this example, however, curious readers can read these details at [47].

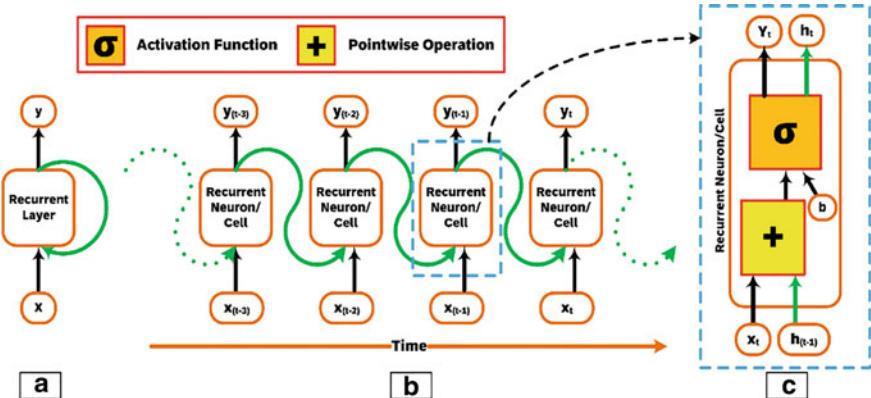


Fig. 6 Schematic of **a** a simple standard recurrent layer, **b** unrolled version of a simple recurrent layer, **c** detailed components of a recurrent neuron or cell

Although RNN is powerful in using recurrent information to predict the future, unfortunately, when there is a long gap between sequential data, RNN architecture forgets its previous knowledge. As such, the Long Short Term Memory (LSTM) network has been introduced in [49] as a new variation of RNN networks to solve this problem. Figure 7 shows main components of one standard LSTM module. In each LSTM module information from previous layer and from the current layer pass through few sub-modules/gates, each of which is dedicated to a specific task. In general, an LSTM module aims to act as a memory and predict next time step.

In LSTM architecture, the key idea is that the network can learn what information to store in the long-term state, what to throw away (i.e., forget), and what to read from

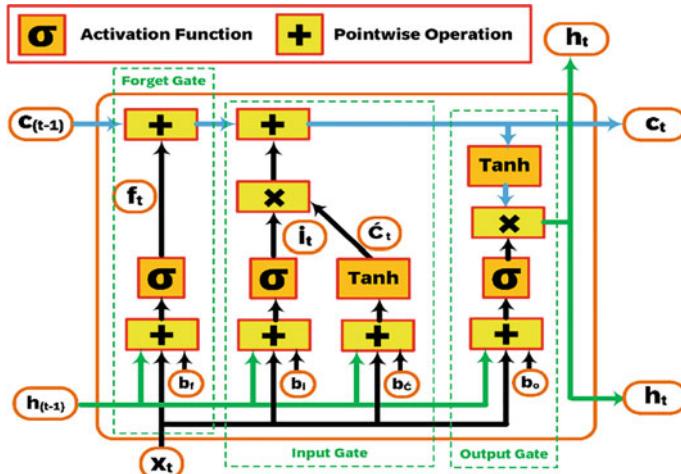


Fig. 7 Schematic of LSTM architecture with forget gate

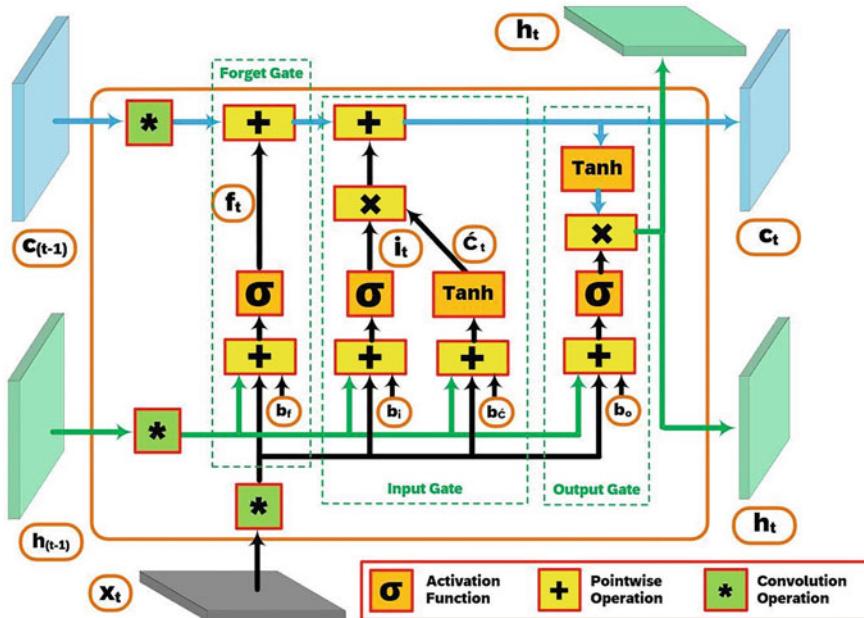


Fig. 8 Details of the recurrent structure of ConvLSTM module used in ConvLSTM network

it. In Figs. 7 and 8, c is a parameter called state in time which is in charge of keeping long-term state of data [47]. It traverses the network from left to right, and it first goes through a forget gate, dropping some memory, and then it adds some new memory by passing through input gate. The result is copied, as one-part passes to the output updating long-term memory, and the copied version passes through the output gate to produce the short-term state. As such, using the forget, input, and output gates, LSTM learns which part of the input should be preserved for a long-term and which part of information should be erased.

Although one LSTM module alone is powerful to be used for small-scale datasets, there is a limited capacity of that in handling engineering problems. For this reason, the LSTM cells have to be arranged into a specific network architecture when processing practical data. In this way, LSTM network can comprise of several consecutive LSTM modules and also other types of layers in between. Furthermore, since these networks are designed for text analysis, the number of input dimensions is one, so they are capable of handling one-dimensional data types, such as text, signals, etc., not suitable for spatial sequence data such as video and image sequence data. For fire image sequence data, there would be four dimensions: two for spatial coordinates of pixels, one dimension for color, brightness, or IR radiation, and one dimension for frame number or time.

To adopt LSTM networks for vision data analyses, ConvLSTM is proposed in [47] which is a combination of Deep Convolutional Neural Network (DCNN) [2] and LSTM [49] module. One ConvLSTM module is illustrated in Fig. 8. Activation

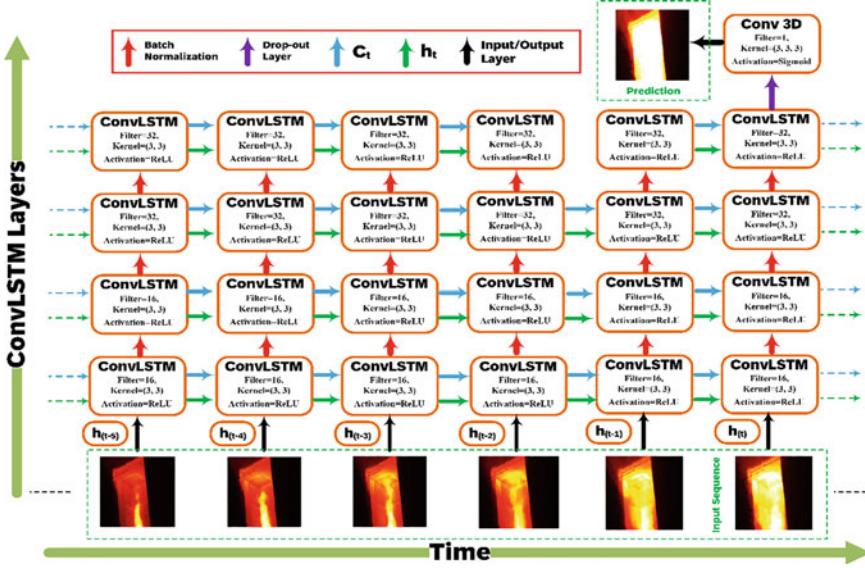


Fig. 9 The architecture of the ConvLSTM used in this study. The details of the network explained in the legend with colours. For the sake of presentation, only few loops of the network illustrated

functions (i.e., sigmoid function, σ , and tanh function), between layers add non-linearity to the model for better generalization of the data. As can be seen from the Fig. 8, in a ConvLSTM module, matrix multiplication at each gate of classical LSTM is replaced with Convolutional operation. Now a ConvLSTM network as depicted in Fig. 9 has several ConvLSTM modules, and the information flow has two paths, one vertical through layers of the network, and one horizontal (recurrent) through time. The last layer in ConvLSTM network is a 3D Convolutional layer with a sigmoid activation function to provide future prediction.

In this example, to train ConvLSTM model, different datasets were collected from actual room fire tests. Thermal IR video frames recorded by a FLIR T650sc camera from a set of full-scale room fire tests were used. The details of the room set-ups and fire tests are provided in [50]. IR video data were first re-scaled into smaller image size. The frame-rate of each video was calculated, and one frame for every second was retained. The entire dataset was normalized then by Min–Max normalization. In general, the dataset we used in experiments had a tensor of length $3338 \times 20 \times 64 \times 64$ (i.e., 3338 Gy-scale videos of 20 s with size of 64×64). The dataset was randomly shuffled and separated into 90%, 5%, and 5% for training, validation, and test purposes, respectively. Figure 10 presents two example data from the dataset. Even in these examples, it can be seen how fast and dangerous a room fire can be.

Figures 11 and 12 shows some prediction examples for a qualitative comparison of ConvLSTM in prediction of rapid fire growth. In these figures, the top row is ten second past frames and the second row is ten second frames of the future (i.e., ground truth frames). The third row is the prediction results of ConvLSTM

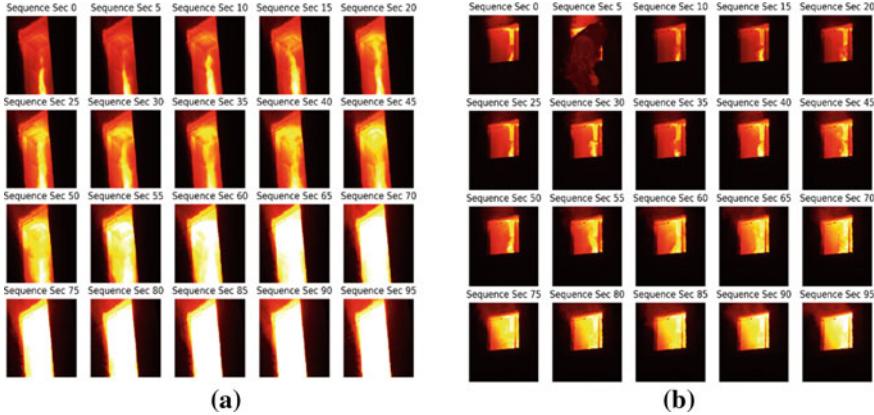


Fig. 10 Two randomly selected data sample from the entire dataset. For the sake of illustration, we show every 5 s

for ten second in future. Comparing predicted frames with ground truth frames, we observe that ConvLSTM generate predictions that are aligned with the ground-truth frames. Flame behaviors is of interest in fire safety because its characteristics affect fire spread. The predicted fire growth in ten seconds by ConvLSTM are shown in Figs. 11 and 12 (with yellow colors, corresponding to higher temperatures), which demonstrate a positive correlation with ground truth frames. The results indicate that the ConvLSTM model learns general shape of fire plumes and its movements and more importantly the growth rate. One interesting achievement was that ConvLSTM could provide prediction even for occluded frames for example by moving personals in front of the camera (see one example in Fig. 10b). Results of ConvLSTM shows that this model can generate future frames with the general and abstract details learned from the past. The predicted frames compare well with the corresponding GT images, yet with some discrepancy.

To further analyze the results of ConvLSTM, every three second of the predicted futures along with their corresponding ground truth images are illustrated in Fig. 12.

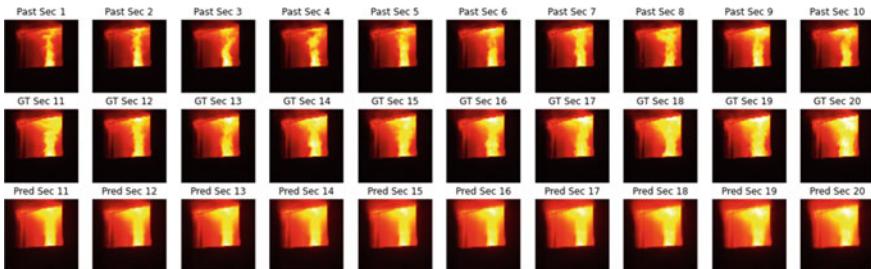


Fig. 11 Results of ConvLSTM applied on a selected test data (For more information regarding the test set-ups, please see ([50] for test #9)

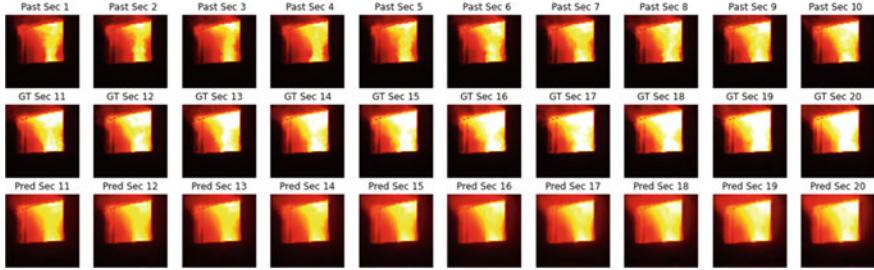


Fig. 12 Results of ConvLSTM applied on a selected test data (For more information regarding the test set-ups, please see ([50] for test #10)

In this figure, we reported the colormap of each image which shows the temperature range distribution. By looking at the figure, temperature differences between ground truth and predicted images is slightly different. Generating temperature is significantly difficult since it depends on many factors, especially stage of the fire in which temperature of the room might be growing or cooling. ConvLSTM prediction is not perfect, but it could predict the temperature trend direction. Note that the colour map shows average temperature (i.e., histogram) of colours in this figure, and losing details caused the temperature differences between predicted and GT images (Fig. 13).

The experimental results also revealed a predictable and trivial conclusion that the further the future, the harder the anticipation. The graph in Fig. 14 shows the variation between prediction of one second in future by ConvLSTM model for different all test videos used in this example. Note that these videos are unseen and captured by almost different test setup (e.g., various opening specifications, camera locations, angle of view, temperatures, and fire behavior). From Fig. 14, the maximum range of error and similarity differences are negligible, and the performance of the model for various data situations is significantly stable. The structural similarity index measure (SSIM), the Mean Squared Error (MSE), the Root Mean Squared Error (RMSE), and the Mean Absolute Error (MAE) are reported in the figure [51].

4 Future of Generative AI in Fire Safety

The future of Generative AI in fire safety holds great potential for enhancing prevention, response, and recovery efforts for fire safety applications. One of the main reasons for the advancement of AI is the availability of large digital databases in different fields of science and technologies. However, this is not always the case since in fire applications data collection is usually costly, dangerous, and labor-intensive. Generative AI can be an alternative for these challenging situations where lack of data is a hurdle for adaptation of AI as a solution.

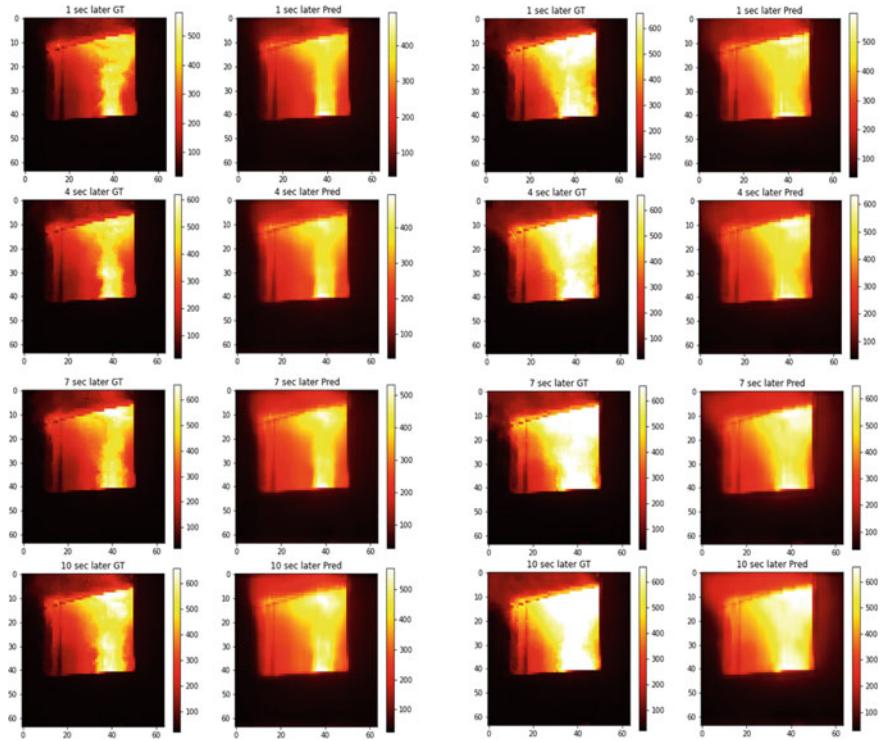


Fig. 13 Selected frames from Fig. 10 (left image) and Fig. 11 (right image) for better illustration of prediction details by ConvLSTM model. Temperature colormaps are also presented for better comparison

Generative AI can be utilized to create realistic fire simulations for training purposes. By generating virtual fire scenarios, AI models can help firefighters and emergency responders practice their skills in a safe and controlled environment, enabling them to learn effective strategies and improve their decision-making abilities. These AI models can analyze historical fire data, weather patterns, building structures, and other relevant factors to predict the likelihood of fires occurring in specific areas. By identifying high-risk regions, authorities can prioritize their fire prevention measures, allocate resources efficiently, and take proactive steps to mitigate potential hazards. Benefits of Generative AI are endless, and these models can be trained on large datasets of fire-related information, including thermal imaging, smoke patterns, and sensor data, to detect fires in their early stages. This can enable the development of advanced fire detection systems that provide real-time alerts, allowing for quicker response times and minimizing potential damage. As a more futuristic application, Generative AI can assist in devising optimal evacuation plans during fire emergencies. By considering factors such as building layouts, population

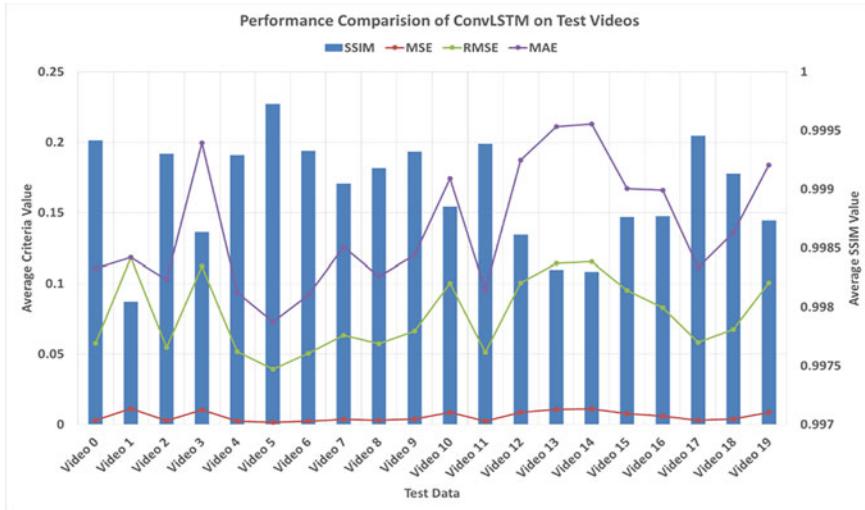


Fig. 14 Results of comparison study on performance of ConvLSTM in prediction of one frame in future for room fire test IR dataset

density, and traffic patterns, AI models can generate evacuation routes that maximize efficiency and minimize congestion, ensuring the safe and swift evacuation of affected individuals.

Robotics and smart devices are becoming more common for fire safety application, and Generative AI can play a significant role in the development of firefighting robots. These robots could be equipped with advanced sensors, cameras, and AI algorithms to navigate through hazardous environments, locate and suppress fires, and assist firefighters in their operations. Generative AI can contribute to the training of such robots by simulating various fire scenarios and enabling the robots to learn effective firefighting strategies. After a fire incident, Generative AI can aid in the reconstruction and recovery process. By analyzing data from damaged structures, AI models can generate 3D models and simulations that assist architects, engineers, and construction teams in planning and executing the restoration efforts effectively.

Based on many advantages that Generative AI models have, we anticipate that there will be more examples of these AI architectures in future, helping AI researchers and developers to create new solutions for fire safety problems.

5 Conclusion

In this chapter, we discussed general idea of Generative AI models. Generative AI refers to a branch of artificial intelligence that focuses on creating and generating new content, such as images, videos, text, and even music. It uses deep learning models,

particularly generative models like Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), to learn patterns from existing data and generate new, original content that resembles the training examples. These models are special neural networks designed to show how data are generated and how we can extract important features from available data to generate new similar data. We showed that Generative AI have been using in many fields for various purposes, and fire safety is not an exception. In this chapter, after reviewing examples of using Generative AI for fire safety from the literature, with more details, we described two recent successful examples of Generative AI, one for translating RGB vision data of fire into thermal IR vision data without using any IR camera, and the other example was how Generative AI can predict fire growth rate in future visually.

Generative AI has gained significant attention due to its ability to generate realistic and creative outputs. It has been used in various fields, including art, entertainment, design, and even scientific research. We explained some popular applications of Generative AI including generating realistic images, creating new visuals that resemble existing ones or even produce entirely new and imaginative content. These models can generate new video sequences based on existing footage or create entirely synthetic videos. Generative AI has the potential to revolutionize many industries by enabling automated content creation, enhancing creativity, and assisting in complex decision-making processes. we mentioned benefits of Generative AI, however, it's important to note that while Generative AI holds promise in fire safety, its implementation should be accompanied by rigorous testing, adherence to safety standards, and continuous human oversight. Collaborations between AI experts, fire safety professionals, and regulatory bodies will be essential in shaping the future of Generative AI in this field.

References

1. Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., & Tang, X. (2017). *Residual attention network for image classification* (pp. 3156–3164). https://openaccess.thecvf.com/content_cvpr_2017/html/Wang_Residual_Attention_Network_CVPR_2017_paper.html
2. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), Article 7553. <https://doi.org/10.1038/nature14539>
3. Rajoub, B. (2020). Chapter 3—Supervised and unsupervised learning. In W. Zgallai (Ed.), *Biomedical signal processing and artificial intelligence in healthcare* (pp. 51–89). Academic Press. <https://doi.org/10.1016/B978-0-12-818946-7.00003-2>
4. Mozaffari, M. H. (2020). *A real-time and automatic ultrasound-enhanced multimodal second language training system: A deep learning approach* [Ph.D. Thesis, Université d'Ottawa/University of Ottawa]. <https://doi.org/10.20381/ruor-24710>
5. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
6. Eddy, S. R. (1996). Hidden Markov models. *Current Opinion in Structural Biology*, 6(3), 361–365. [https://doi.org/10.1016/S0959-440X\(96\)80056-X](https://doi.org/10.1016/S0959-440X(96)80056-X)
7. Jiang, L., Zhang, H., & Cai, Z. (2009). A Novel Bayes model: Hidden Naive Bayes. *IEEE Transactions on Knowledge and Data Engineering*, 21(10), 1361–1371. <https://doi.org/10.1109/TKDE.2008.234>

8. Viroli, C., & McLachlan, G. J. (2019). Deep Gaussian mixture models. *Statistics and Computing*, 29(1), 43–51. <https://doi.org/10.1007/s11222-017-9793-z>
9. Zhang, N., Ding, S., Zhang, J., & Xue, Y. (2018). An overview on restricted Boltzmann machines. *Neurocomputing*, 275, 1186–1199. <https://doi.org/10.1016/j.neucom.2017.09.065>
10. Hinton, G. (2009). Deep belief networks. *Scholarpedia*, 4(5), 5947. <https://doi.org/10.4249/scholarpedia.5947>
11. Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1), 53–65. <https://doi.org/10.1109/MSP.2017.2765202>
12. Zhai, J., Zhang, S., Chen, J., & He, Q. (2018). Autoencoder and its various variants. In *2018 IEEE international conference on systems, man, and cybernetics (SMC)*, (pp. 415–419). <https://doi.org/10.1109/SMC.2018.00080>
13. Wong, C. S., & Li, W. K. (2000). On a mixture autoregressive model. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(1), 95–115. <https://doi.org/10.1111/1467-9868.00222>
14. Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11), 2673–2681. <https://doi.org/10.1109/78.650093>
15. Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. ArXiv Preprint [arXiv:1312.6114](https://arxiv.org/abs/1312.6114)
16. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Lukasz, & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
17. Gozalo-Brizuela, R., & Garrido-Merchán, E. C. (2023). *A survey of generative AI applications* ([arXiv:2306.02781](https://arxiv.org/abs/2306.02781)). arXiv. <https://doi.org/10.48550/arXiv.2306.02781>
18. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139–144.
19. Gillioz, A., Casas, J., Mugellini, E., & Abou Khaled, O. (2020). Overview of the transformer-based models for NLP Tasks. In *2020 15th Conference on computer science and information systems (FedCSIS)* (pp. 179–183).
20. Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, (pp. 2223–2232).
21. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., & Funtowicz, M. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: System demonstrations*, (pp. 38–45).
22. Shaw, P., Uszkoreit, J., & Vaswani, A. (2018). Self-attention with relative position representations. ArXiv Preprint [arXiv:1803.02155](https://arxiv.org/abs/1803.02155)
23. Oussidi, A., & Elhassouny, A. (2018). Deep generative models: Survey. *International Conference on Intelligent Systems and Computer Vision (ISCV), 2018*, 1–8.
24. Zhou, L., Cai, C., Gao, Y., Su, S., & Wu, J. (2018). Variational autoencoder for low bit-rate image compression. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, (pp. 2617–2620).
25. Spinner, T., Körner, J., Görtler, J., & Deussen, O. (2018). Towards an interpretable latent space: An intuitive comparison of autoencoders with variational autoencoders. *IEEE VIS 2018*.
26. Ko, Y., Mozaffari, M. H., & Li, Y. (2023). Fire and smoke image recognition. In *Intelligent building fire safety and smart firefighting*. Springer.
27. Dung, N. Q., & Kim, H. (2022). Generating high-resolution fire images with controllable attributes via generative adversarial networks. In *2022 22nd International conference on control, automation and systems (ICCAS)* (pp. 348–353). <https://doi.org/10.23919/ICCAS5662.2022.10003687>
28. Qin, K., Hou, X., Yan, Z., Zhou, F., & Bu, L. (2022). FGL-GAN: Global-local mask generative adversarial network for flame image composition. *Sensors*, 22(17), Article 17. <https://doi.org/10.3390/s22176332>

29. Yun, K., Bustos, J., & Lu, T. (2018). *Predicting rapid fire growth (Flashover) using conditional generative adversarial networks* (arXiv:1801.09804). arXiv. <https://doi.org/10.48550/arXiv.1801.09804>
30. Kacker, T., Perrusquia, A., & Guo, W. (2023). Multi-spectral fusion using generative adversarial networks for UAV detection of wild fires. In *2023 International conference on artificial intelligence in information and communication (ICAIIC)* (pp. 182–187). <https://doi.org/10.1109/ICAIIC57133.2023.10067042>
31. Park, M., Tran, D. Q., Bak, J., & Park, S. (2022). Advanced wildfire detection using generative adversarial network-based augmented datasets and weakly supervised object localization. *International Journal of Applied Earth Observation and Geoinformation*, 114, 103052. <https://doi.org/10.1016/j.jag.2022.103052>
32. Cheng, S., Ma, J., & Zhang, S. (2019). Smoke detection and trend prediction method based on Deeplabv3+ and generative adversarial network. *Journal of Electronic Imaging*, 28(3), 033006. <https://doi.org/10.1117/1.JEI.28.3.033006>
33. Geetha, S., Abhishek, C. S., & Akshayanaat, C. S. (2021). Machine vision based fire detection techniques: A survey. *Fire Technology*, 57(2), 591–623. <https://doi.org/10.1007/s10694-020-01064-z>
34. Chaturvedi, S., Khanna, P., & Ojha, A. (2022). A survey on vision-based outdoor smoke detection techniques for environmental safety. *ISPRS Journal of Photogrammetry and Remote Sensing*, 185, 158–187. <https://doi.org/10.1016/j.isprsjprs.2022.01.013>
35. Fahrmeir, L., Kneib, T., Lang, S., & Marx, B. D. (2022). Regression models. In *Regression: Models, methods and applications* (pp. 23–84). Springer.
36. Mozaffari, M., Li, Y., & Ko, Y. (2022). Detecting flashover in a room fire based on the sequence of thermal infrared images using convolutional neural networks. *Proceedings of the Canadian conference on artificial intelligence*.
37. Kuang, X., Zhu, J., Sui, X., Liu, Y., Liu, C., Chen, Q., & Gu, G. (2020). Thermal infrared colorization via conditional generative adversarial network. *Infrared Physics and Technology*, 107, 103338. <https://doi.org/10.1016/j.infrared.2020.103338>
38. Li, Y., Ko, Y., & Lee, W. (2022). RGB image-based hybrid model for automatic prediction of flashover in compartment fires. *Fire Safety Journal*, 132, 103629. <https://doi.org/10.1016/j.firesaf.2022.103629>
39. Yi, S., Li, J., & Yuan, X. (2021). DFPGAN: Dual fusion path generative adversarial network for infrared and visible image fusion. *Infrared Physics and Technology*, 119, 103947. <https://doi.org/10.1016/j.infrared.2021.103947>
40. Wang, Z., Chen, J., & Hoi, S. C. H. (2021). Deep learning for image super-resolution: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10), 3365–3387. <https://doi.org/10.1109/TPAMI.2020.2982166>
41. Jing, Y., Yang, Y., Feng, Z., Ye, J., Yu, Y., & Song, M. (2020). Neural style transfer: A review. *IEEE Transactions on Visualization and Computer Graphics*, 26(11), 3365–3385. <https://doi.org/10.1109/TVCG.2019.2921336>
42. Chu, C., Zhmoginov, A., & Sandler, M. (2017). Cyclegan, a master of steganography. ArXiv Preprint [arXiv:1712.02950](https://arxiv.org/abs/1712.02950)
43. Li, Y. (2021). *Dual-attention generative adversarial network and flame and smoke analysis* [Master's Thesis]. Université d'Ottawa/University of Ottawa.
44. Bawalya, A., Gibbs, E., Lougheed, G., & Kashef, A. (2023, June 7). *Characterization of fires in multi-suite residential dwellings: Final project report: Part 1-A compilation of post-flashover room fire test data—NRC Publications Archive*. <https://nrc-publications.canada.ca/eng/view/object/?id=a46dbb8a-5093-4d5e-b5d3-07bbc866fd9a>
45. Li, Y., Lee, W., & Ko, Y. (2023). A feasibility study on translation of RGB images to thermal images: Development of a machine learning algorithm. *Springer Nature SN Computer Science*. Under Publication.
46. Sun, Y., Wang, Y., Hu, L., Huang, Y., Liu, H., Wang, S., & Zhang, C. (2023). Attribute-Guided generative adversarial network with improved episode training strategy for few-shot SAR image generation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 16, 1785–1801.

47. Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., & Woo, W. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in Neural Information Processing Systems*, 28.
48. Mikolov, T., Karafiat, M., Burget, L., Cernocky, J., & Khudanpur, S. (2010). Recurrent neural network based language model. *Interspeech*, 2(3), 1045–1048.
49. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
50. Mozaffari, M. H., Li, Y., & Ko, Y. (2023). Real-time detection and forecast of flashovers by the visual room fire features using deep convolutional neural networks. *Journal of Building Engineering*, 64, 105674.
51. Ndajah, P., Kikuchi, H., Yukawa, M., Watanabe, H., & Muramatsu, S. (2010). SSIM image quality metric for denoised images. In *International conference on visualization, imaging and simulation—Proceedings* (p. 57).

A Multi-scale Convolutional Autoencoder with Attention Mechanism for Fault Diagnosis of Rotating Machinery



Zihao Lei, Hongguang Yun, Feiyu Tian, Guangrui Wen, and Zheng Liu

Abstract In recent years, intelligent fault diagnosis based on deep learning has achieved vigorous development due to its powerful feature representation ability. However, the data collected from industrial sites often contain different levels of noise, which makes it difficult to extract effective fault features, which seriously affects the performance of the model. In addition, the limited labeled data makes the training of deep network models even more challenging. To address the above problems, a multi-scale convolutional autoencoder with attention mechanism (MSCAE-AM) is developed. Specifically, as a typical unsupervised learning model, the encoder can effectively reduce the dependence on labeled data. Furthermore, the feature extraction ability of the model in noisy environments can be improved by combining noise reduction operations and embedding multi-scale convolutional layers and attention mechanisms. Experimental results on the wind turbine fault simulation datasets verify the effectiveness and superiority of the proposed method. The results show that the proposed method can not only effectively reduce the dependence on label data, but also has stronger robustness to noise than other methods.

Keywords Rotating machinery · Convolutional autoencoder · Attention mechanism · Fault diagnosis · Interpretability

1 Introduction

With the rapid development of Industry 4.0, modern manufacturing systems have been upgraded to the level of intelligence and robotization. Large rotating machinery plays an indispensable role in modern manufacturing systems and is extensively used

Z. Lei · F. Tian · G. Wen

National Key Lab of Aerospace Power System and Plasma Technology, Xi'an Jiaotong University, Xi'an 710049, China
e-mail: zihao_lei@163.com

Z. Lei · H. Yun · Z. Liu (✉)

School of Engineering, The University of British Columbia, Kelowna, BC V1V 1V7, Canada
e-mail: zheng.liu@ubc.ca

in various industrial fields such as energy, transportation, chemical, aerospace, and electric power [1–3]. Nevertheless, the long-term harsh operating environment makes rotating machinery prone to failure, causing unexpected economic losses and human casualties. Therefore, fault diagnosis is crucial to keep the safe and stable operation of rotating machinery and prevent substantial economic losses [4–6].

Recently, with the development of artificial intelligence technology, Intelligent fault diagnosis (IFD) has become a research hotspot in the field of mechanical fault diagnosis. Intelligent fault diagnosis can be roughly divided into two categories: diagnostic methods based on traditional machine learning and diagnostic methods based on deep learning [7]. Traditional machine learning algorithms, such as KNN, SVM, etc., rely on manually extracted features as input, so there are problems such as poor generalization, and it is difficult to cope with the massive heterogeneous data generated by mechanical equipment under the background of industrial big data. Deep learning (DL) can automatically learn feature extraction from data, which is expected to make up for the inherent defects of traditional intelligent diagnosis methods and build an end-to-end diagnosis model. Deep learning technology has become the most promising means to realize intelligent fault diagnosis because of its good generalization performance and powerful feature extraction ability. Many scholars have tried to apply deep learning models such as CNN, AE, RNN, GAN, Transformer, and their variants to the fault diagnosis of rotating machinery parts, and achieved some success [8–12], improving the accuracy and reliability of diagnosis.

However, due to the complexity of the industrial site environment, the collected data often have varying degrees of noise, making it difficult to extract effective and robust features. Many scholars have researched this issue. Zhang et al. proposed a Deep Convolutional Neural Network with Wide First-layer Kernels, which performs well under noisy environmental conditions [13]. Wang et al. proposed a CNN network model guided by the attention mechanism and adopted a joint learning strategy, which has good noise reduction ability [14]. Xu et al. developed a global contextual multi-scale fusion network, which aims to explore robust features and filter out irrelevant information [15]. Li et al. designed a novel multibranch CNN named IFD-MDCN, including a multiscale denoising branch to extract multi-level information and reduce noise impact [16].

Although the above methods have made some progress, they ignore the following issues: (1) It is difficult to obtain sufficient labeled data at the industrial site for the training of intelligent fault diagnosis models. (2) Most fault diagnosis methods ignore the inherent multi-scale characteristics of the data itself and features at a single scale may not be able to fully represent potential fault information.

Based on the above-mentioned issues, a multi-scale convolutional autoencoder based on attention mechanism is proposed. As a typical unsupervised learning model, autoencoders can not only effectively reduce the dependence on labeled data, but also have certain noise reduction capabilities. Therefore, multi-scale convolutional layers are embedded based on traditional convolutional autoencoders to achieve multi-scale feature extraction. Then, the operation of feature splicing realizes preliminary feature fusion, and the channel attention module is applied to extract useful information for classification from the fused features, thereby further improving the diagnostic

performance of the model. Afterward, the dimensions of the features are compressed through a series of small convolution kernels to extract deep features. Finally, the feature map is sent to a classifier composed of fully connected layers to complete the fault diagnosis task. Experimental results on the wind turbine fault simulation datasets verify the effectiveness and superiority of the proposed method.

The main contributions of this paper are concluded as follows.

- (1) A multi-scale convolutional autoencoder based on attention mechanism is proposed, which can not only effectively reduce the dependence on labeled data but also have good noise robustness.
- (2) The combination of attention mechanism and information entropy improves the interpretability of the model.
- (3) Massive experiments under different noise backgrounds are carried out to verify the effectiveness and superiority of the proposed method.

The rest of this paper is organized as follows: Sect. 2 presents the preliminaries; Sect. 3 introduces the details of the proposed method; Sect. 4 implements case studies to verify the effectiveness and superiority of the proposed method; and the last section draws some conclusions.

2 Preliminaries

2.1 Convolutional Neural Network

Convolutional Neural Network (CNN) has been widely used in object detection, pattern recognition, and other fields because of their powerful feature extraction ability [17]. Furthermore, the convolution operation used in the convolution layer gives the network the “weight sharing” characteristics, which significantly reduces the overall parameter amount of the model, thereby effectively suppressing the overfitting phenomenon.

One-dimensional convolution refers to the convolution operation in which the sliding direction of the convolution kernel is one-dimensional. Figure 1 shows the operation flow of one-dimensional convolution, in which three convolution kernels are used to process the original data with three channels, and the depth of the convolution kernel is usually the same as the original signal.

2.2 Autoencoder

Autoencoder is a typical unsupervised learning algorithm, which generally includes two parts: encoder and decoder. It compresses the original input data through the

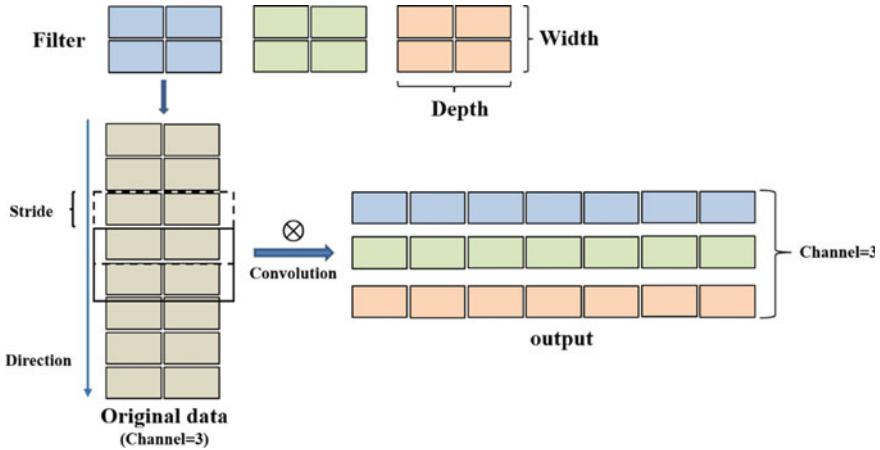


Fig. 1 1-dimensional CNN operation

encoder, maps the compressed data from low-dimensional to high-dimensional through the decoder, and finally realizes the reconstruction of the original data [18].

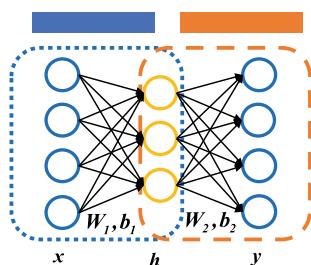
The structure of a traditional autoencoder is shown in Fig. 2. What's more, the structure of the autoencoder is generally symmetrical, and the number of hidden layers in the encoding stage is consistent with that in the decoding stage, and the ultimate goal is to achieve data reconstruction and make the input and output structures the same. The specific calculation process is shown in Eqs. (1) and (2).

$$h_1 = f_e(W_1x + b_1) \quad (1)$$

$$y = f_d(W_2h_1 + b_2) \quad (2)$$

where W_1 and b_1 are the weight and bias of the encoder respectively, W_2 and b_2 are the weight and bias of the decoder respectively, f_e is the activation function of the encoding process, and f_d is the activation function of the decoding process.

Fig. 2 The traditional structure of AE



In the autoencoder, the L2-norm of the input and output is generally used as the loss function, to minimize the error of the input and output during the training process. The specific expression is shown in Eq. (3).

$$L_{L2} = \frac{1}{N} \sum_{i=1}^N \|x^{(i)} - y^{(i)}\|_2^2 \quad (3)$$

where L_{L2} is the error function, N is the number of samples, $x^{(i)}$ and $y^{(i)}$ represents the input and output of the i -th sample respectively.

Furthermore, the weights and biases in the autoencoder are obtained by minimizing an objective function [19]. Besides the L2 norm, cross-entropy is also often adopted as a loss function, which can be shown in Eq. (4).

$$L_{CE} = - \sum_{i=1}^N (x_i \log(y_i) + (1 - x_i) \log(1 - y_i)) \quad (4)$$

According to the above, it can be obtained that the calculation of the above loss function is only related to the data itself, and does not involve label information. Therefore, the autoencoder is a typical unsupervised learning method.

2.3 Attention Mechanism

The feature maps representing the extracted information are output by each layer in the deep neural network. By default, the importance of each feature map is generally the same, but its contribution to the final result varies. Since it is difficult to manually select the feature maps, it is necessary to explore a method that can automatically extract the features of interest.

Humans can focus on some areas of interest when looking at a scene, which avoids the interference of irrelevant information and thus increases the efficiency of information processing. Motivated by human attentional behavior, Vaswani A et al. developed the Transformer structure, which has been a great success in the field of NLP and set off a research boom in AM [20].

In the field of deep learning, AM concentrates the current task on more features of interest by allocating different weights to each feature map. In general, AM is divided into spatial attention mechanism and channel attention mechanism. Among them, the channel attention mechanism assigns weights by measuring the importance of each channel to adaptively extract the features of interest.

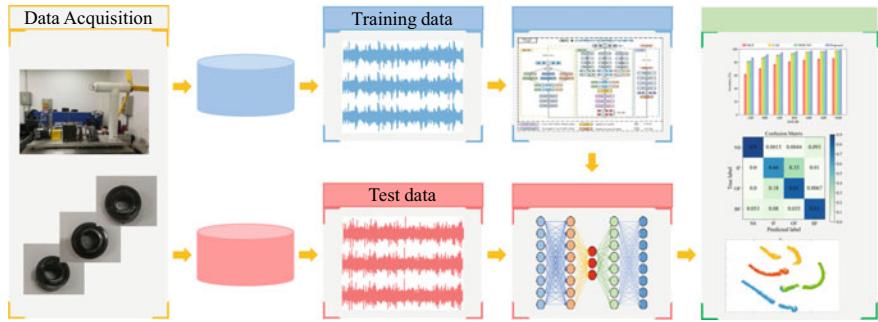


Fig. 3 The flowchart of the proposed method

3 Proposed Method

3.1 The Flowchart of the Proposed Method

The flow chart of the proposed method is presented in Fig. 3, in which four main steps are included:

- Step 1: Acquire raw data from the wind turbine fault simulation platform with the help of vibration sensors.
- Step 2: After the sampled data is preprocessed, the training data and test data are divided according to the ratio of 7:3.
- Step 3: Build the MSCAE-AM model and initialize the model parameters, then train the autoencoder until the loss function converges. Next, fine-tune the encoder and classifier until the model converges, and save the parameters of the encoder and classifier.
- Step 4: Input the unlabeled test data set into the saved model for the fault diagnosis task.

3.2 The Structure of the MSCAE-AM Model

In this paper, a multi-scale convolutional autoencoder model based on the channel attention mechanism is developed. Figure 4 presents its specific structure, which mainly includes three components: encoder, decoder, and classifier. The encoder completes the mapping from the original data to the hidden space and extracts the potential feature information. Then, the decoder is used to realize the reconstruction of the original signal, and finally, the classifier is used to realize the final fault diagnosis task.

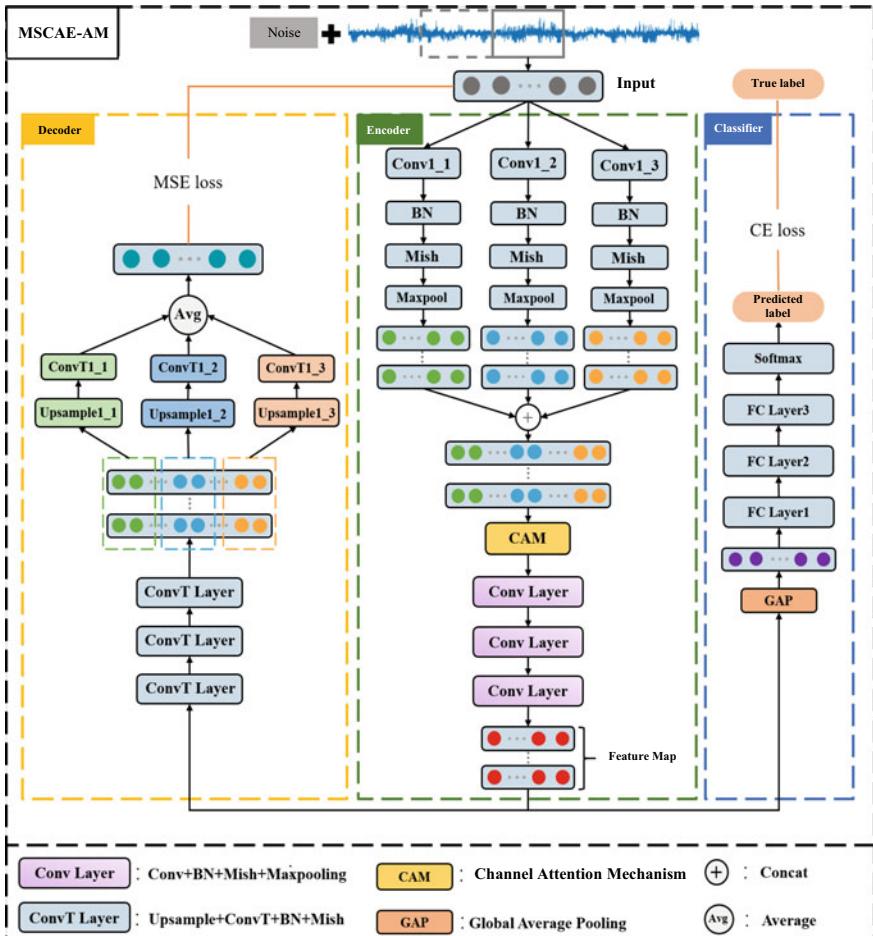
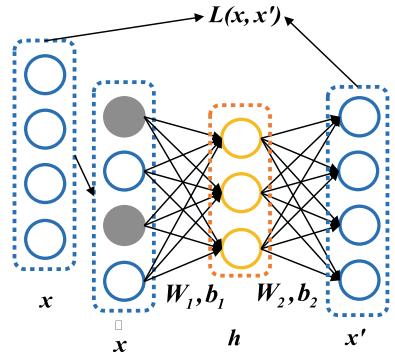


Fig. 4 The structure diagram of the MSCAE-AM model

3.2.1 Denoising Autoencoder

The autoencoder essentially realizes the mandatory compression of the original data through the constraints of the loss function, so that the output of the hidden layer represents the original data. Therefore, the hidden layer's expressiveness determines an autoencoder's performance.

The denoising autoencoder was proposed by Vincent et al. [21] to improve the robustness of the autoencoder. They believe that just as humans can accurately identify partially occluded or partially damaged images, a robust model can still represent the same representation as much as the undamaged signal when the data is partially damaged. As shown in Fig. 5, the core idea of the denoising autoencoder is to take

Fig. 5 The structure of DAE

the noise-added data as input and make the reconstructed data as identical as possible to the original data without noise.

In this paper, based on the above-mentioned ideas, the damaged data is generated by adding noise and input into the autoencoder, and the noise reduction ability of the autoencoder itself is improved through continuous training, thereby enhancing the overall robustness of the model to noise.

3.2.2 Multi-scale Convolution Module

To further enhance the feature extraction capability of the network, the first layer of the autoencoder is designed with multi-scale convolution kernels. The function of the large convolution kernel is similar to that of the short-time Fourier transform. Large convolution kernels of different scales are equivalent to using multiple time windows of different widths, enhancing the network's multi-resolution capability. Among them, the mathematical expression of the convolution operation is as follows.

$$y^{l(i,j)} = K_i^l * X^{l(r^j)} = \sum_{j'}^{W-1} K_i^{l(j')} X^{l(j+j')} \quad (5)$$

where $X^{l(r^j)}$ represents the j -th convolutional area in the l -th layer; W represents the width of the convolution kernel in the convolution layer, and $K_i^{l(j')}$ represents the j -th weight in the i -th convolution kernel in the l -th layer.

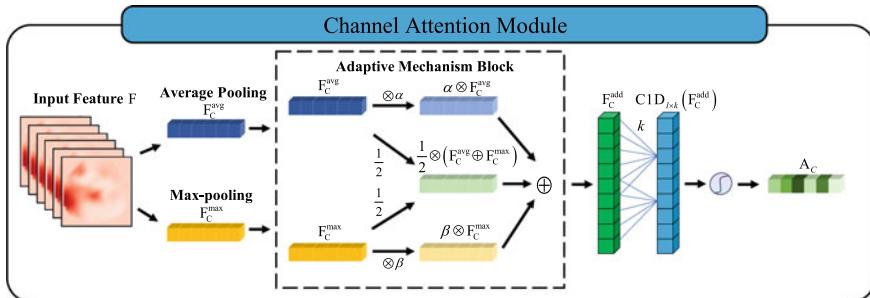
In addition, the network parameter settings are shown in Table 1.

3.2.3 Channel Attention Mechanism Module

In this section, the channel attention mechanism module (CAM) is embedded in the autoencoder to realize the adaptive weighting of each channel of the feature map

Table 1 The network parameter settings for the multi-scale convolution module part

Network structure	Parameter settings
Multi-scale convolution module	$k = 64 * 1; s = 16; p = 24; c = 16$
	$k = 32 * 1; s = 8; p = 12; c = 16$
	$k = 16 * 1; s = 4; p = 6; c = 16$
Deep convolution module	$k = 3 * 1; s = 1; p = 1; c = 32$
	$k = 3 * 1; s = 1; p = 1; c = 64$
	$k = 3 * 1; s = 1; p = 1; c = 64$
Pooling layer	$k = 2 * 1; s = 2$
Activation function	Mish

**Fig. 6** The structure of the channel attention mechanism

extracted by the multi-scale convolution layer so that the model can focus on more useful information for the current task. The specific structure of the channel attention mechanism is shown in Fig. 6.

Furthermore, the feature map is subjected to maximum pooling and average pooling and then sent to the shared MLP. Next, the output results are concatenated. Finally, the activation is completed through the Sigmoid function to generate the channel weight sequence to be assigned. The mathematical form of the above process can be expressed as follows.

$$\begin{aligned} M_c(\mathbf{F}) &= \sigma(MLP(\text{AvgPool}(\mathbf{F})) + MLP(\text{MaxPool}(\mathbf{F}))) \\ &= \sigma(\mathbf{W}_1(\mathbf{W}_0(F_c^{\text{avg}})) + \mathbf{W}_1(\mathbf{W}_0(F_c^{\text{max}}))) \end{aligned} \quad (6)$$

where, \mathbf{F} represent the input features, σ represents the sigmoid activation function, \mathbf{W}_0 and \mathbf{W}_1 represent the weight of MLP.

4 Experimental Verification

In this section, to validate the effectiveness and the superiority of the proposed method, case studies using the wind turbine fault simulation platform are conducted. Moreover, to better illustrate the performance of the proposed method, feature visualization analysis, confusion matrix analysis, and visualization of attention weights are implemented.

To ensure the stability of the training and testing process, the mean-std normalization method is used to normalize the raw data. Sufficient data is the basis for training deep neural networks. Thus, to obtain more samples and enhance the generalization of the networks, the data augment trick, sliding windows processing, is adopted to realize overlap sampling. To the influence of accidental factors, 70% of samples are randomly selected for the training set, and the other 30% of samples are used as the test set.

4.1 Datasets Description

The wind turbine fault simulation datasets are used to further verify the effectiveness and the superiority of the proposed method. The data set of bearing is collected from the wind turbine fault simulator, the composition of which is shown in Fig. 7.

The type of experimental bearing is ER16K. The vibration data of four types of bearings was collected by the accelerometer at 20,480 Hz sampling frequency, including Normal State (NS), Outer Fault (OF), Inner Fault (IF), and Ball Fault (BF). And the faulty bearing is shown in Fig. 8.

The vibration data from 3 directions, including horizontal, vertical, and axial, are collected by Brüel & Kjaer Deltatron accelerometer. In the case study, the vibration data collected from the vertical direction is adopted, and the speed condition is set as

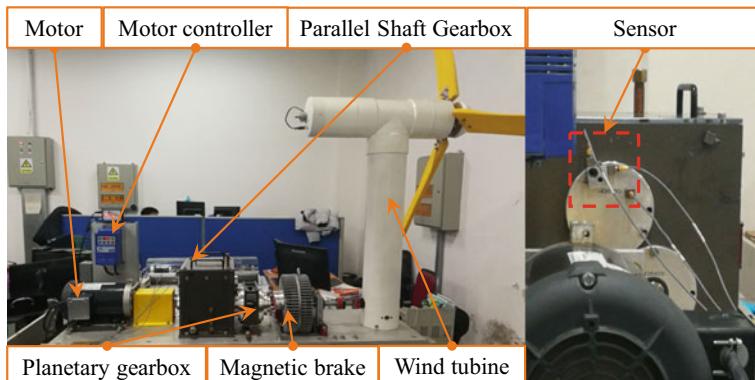


Fig. 7 The experiment platform



Fig. 8 The fault types of bearing

1500 rpm. The number of samples for each health condition is 750 and each sample contains 1024 sampling points.

4.2 Comparison Methods

To further illustrate the superiority of MSCAE-AM, three methods are selected for comparison, including Multilayer Perceptron (MLP), Convolutional Auto-Encoder (CAE), and Deep Convolutional Neural Networks with Wide First-layer Kernel (WDCNN).

(1) MLP

Multilayer Perceptron (MLP, Multilayer Perceptron), also known as Artificial Neural Network (ANN, Artificial Neural Network), is a basic neural network model. This section utilizes a 5-layer MLP embedded with a BN (Batch Normalization) layer.

(2) CAE

To illustrate the effectiveness of the improvement measures in this paper, the traditional convolutional auto-encoder (CAE, Convolutional Auto-Encoder) is selected as one of the comparison methods. Except for the multi-scale convolution layer and the attention mechanism module, the network structure setting of CAE is consistent with the proposed method.

(3) WDCNN

Due to its good robustness, the Deep Convolutional Neural Network with Wide First-layer Kernel Model (WDCNN) is widely used in the comparison of fault diagnosis. In the WDCNN model, except the first layer uses a wide convolution kernel, the rest of the layers are small convolution kernels as 3×1 , the model has many layers, strong expressive ability, and strong robustness to noise.

4.3 Experimental Results and Analysis

4.3.1 Classification Results Under Different Noise Backgrounds

Due to the complexity of the actual industrial site, the acquired signals will be undoubtedly disturbed by noise, and the robustness to noise is an important indicator to measure the performance of the model.

To compare the anti-noise performance of each method, this section injects different intensities of Gaussian noise into the original vibration signal for training and testing of the network. And it can be represented as follows.

$$SNR_{dB} = 10 \log_{10} \left(\frac{P_{\text{signal}}}{P_{\text{noise}}} \right) \quad (7)$$

where, P_{signal} represents the signal power, and P_{noise} represents the noise power.

To avoid the effect of chance factors such as initialization of network parameters, the average of ten training results is taken as the accuracy of each method. Table 2 and Fig. 9 show the specific results. According to the above results, we can find that as the noise intensity increases, the performance of each model will be affected, and the average accuracy will also drop to some extent. In addition, the average accuracy of MLP decreases significantly with the increase in noise intensity. What's more, when the noise intensity is small, the accuracy is still at a relatively low level, which indicates that the basic neural network is easily overfitted due to a large number of parameters, and the pattern recognition task cannot be well completed. When the SNR ratio is set as 10 dB, the noise is at a low level, and the average accuracy of CAE and WDCNN is 97.02% and 97.30% respectively, which is lower than the 98.51% accuracy of the proposed method. Moreover, the gap in the accuracy rate will gradually expand as the SNR ratio decreases, which shows that the method in this paper has strong robustness and can accurately classify faults under a certain intensity of noise interference.

Table 2 The specific results under different levels of noise

Noise/dB	MLP	CAE	WDCNN	Proposed
-2	61.23 ± 0.91	80.72 ± 1.15	81.28 ± 0.91	86.33 ± 0.51
0	69.92 ± 0.72	85.91 ± 1.09	88.06 ± 0.72	91.40 ± 0.42
2	76.38 ± 0.66	90.15 ± 0.99	91.20 ± 0.66	94.18 ± 0.55
4	80.54 ± 0.72	92.93 ± 0.56	93.92 ± 0.72	96.13 ± 0.45
6	82.95 ± 0.43	95.22 ± 0.25	95.19 ± 0.43	97.09 ± 0.30
8	84.76 ± 0.42	96.27 ± 0.36	96.63 ± 0.42	97.87 ± 0.28
10	85.43 ± 0.39	97.2 ± 0.39	97.30 ± 0.39	98.51 ± 0.25

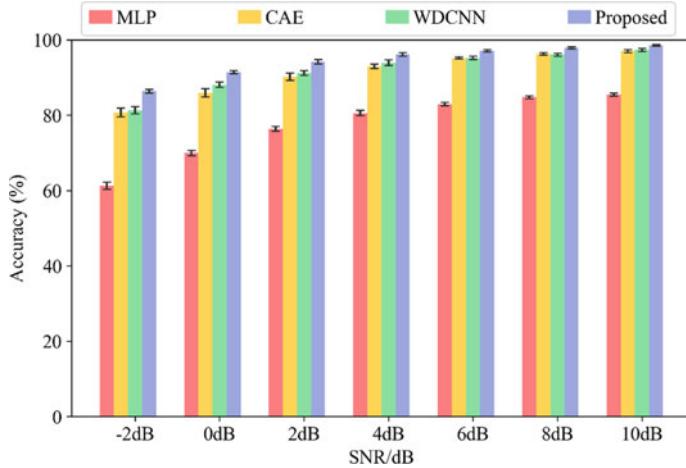


Fig. 9 The results under different levels of noise

4.3.2 Feature Visualization

To demonstrate the features learned by all methods, the feature matrices sent into the classifier are extracted and visualized by the t-distributed stochastic neighbor embedding (t-SNE) approach [22]. We randomly choose the task of $SNR_{dB} = 6$, and the visualized results are shown in Fig. 10. It can be observed that the distribution of the features learned by the proposed method is the best. Besides, the features extracted by MLP are poorly distinguishable. And the features extracted by CAE and WDCNN can realize the distinction of various types of bearing data, but the boundaries of the OF and IF have different degrees of “interweaving”. From the above analysis, it can be seen that compared with other methods, the proposed method can extract more effective features in a certain noise background.

4.3.3 Confusion Matrix Analysis

To show the classification results in detail, under the same task of $SNR_{dB} = 6$, the confusion matrix is drawn. As Fig. 11 shows, the proposed method achieves the best results. In contrast, other methods have different degrees of classification errors. In particular, MLP has an obvious misclassification phenomenon. In addition, in CAE and WDCNN, the phenomenon of misclassification of IF and OF is relatively obvious, which is also consistent with the previous analysis, further proving the superiority of the proposed method.

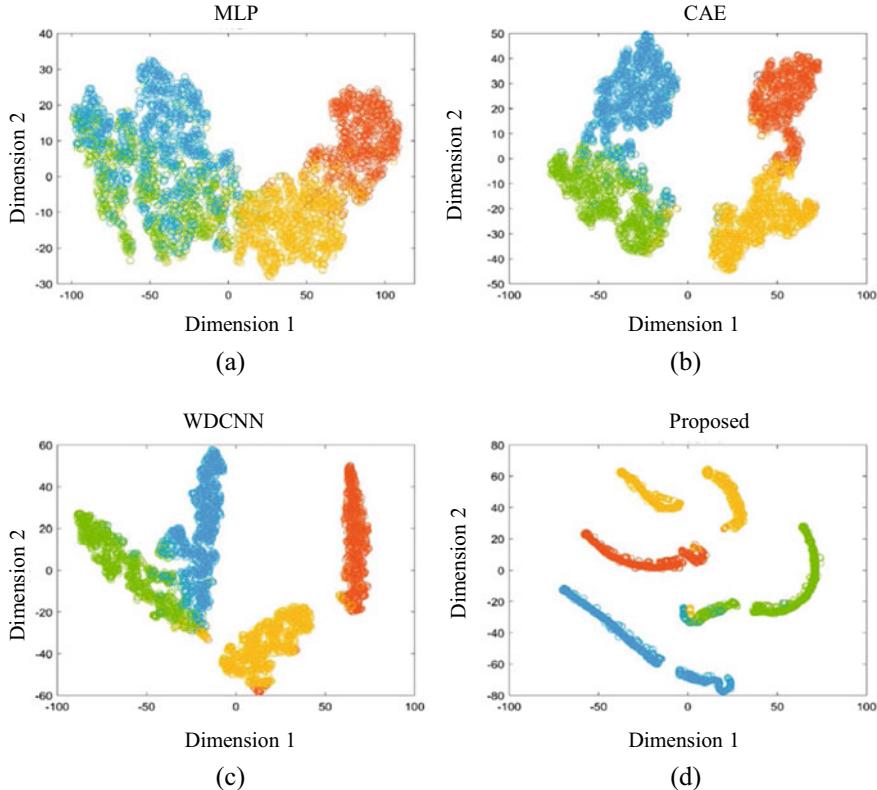


Fig. 10 T-sne visualization results ($SNR_{dB} = 6$). **a** MLP. **b** CAE. **c** WDCNN. **d** Proposed

4.3.4 Visualization of Attention Weights

To further improve the interpretability of the proposed method, the following analysis is carried out. Since the greater the sample entropy of a time series, the higher its complexity and the greater the amount of information contained, it has a higher value at the information level. Based on the above, we visualize the regularization results of the attention weights corresponding to each channel and the sample entropy [23] of the feature map, and the results are shown in Fig. 12. It can be seen from Fig. 12 that the regularized attention weight is positively correlated with the sample entropy of each channel feature map, and the channel attention mechanism tends to give greater weight to feature maps with relatively large sample entropy. This shows that the attention mechanism module in the model can adaptively and effectively weigh and integrate the information extracted by the multi-scale convolutional layer, so that the channel with higher information value contributes more to the final output, thereby improving the overall performance of the model.

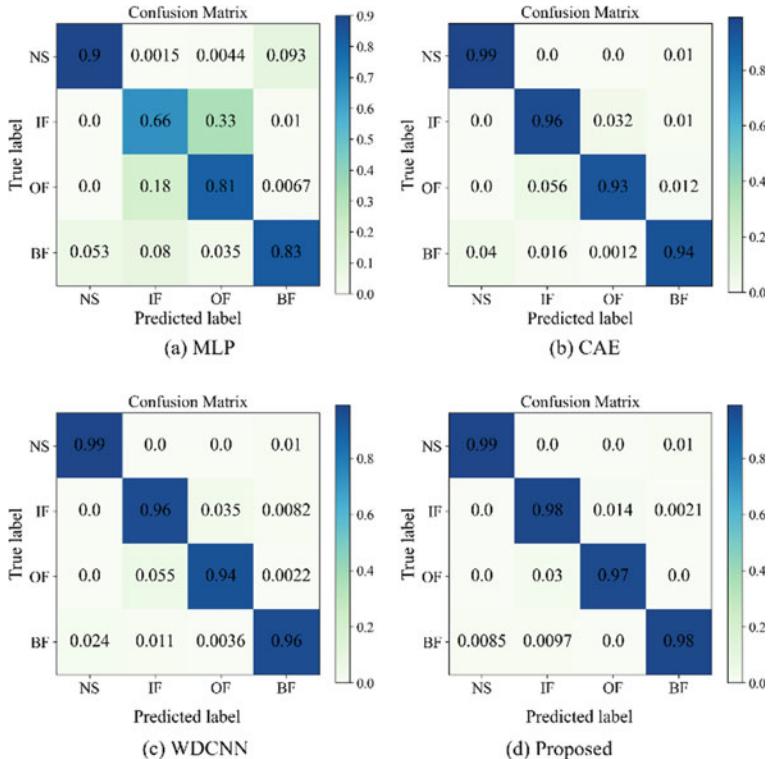


Fig. 11 Confusion matrix ($SNR_{dB} = 6$). **a** MLP. **b** CAE. **c** WDCNN. **d** Proposed

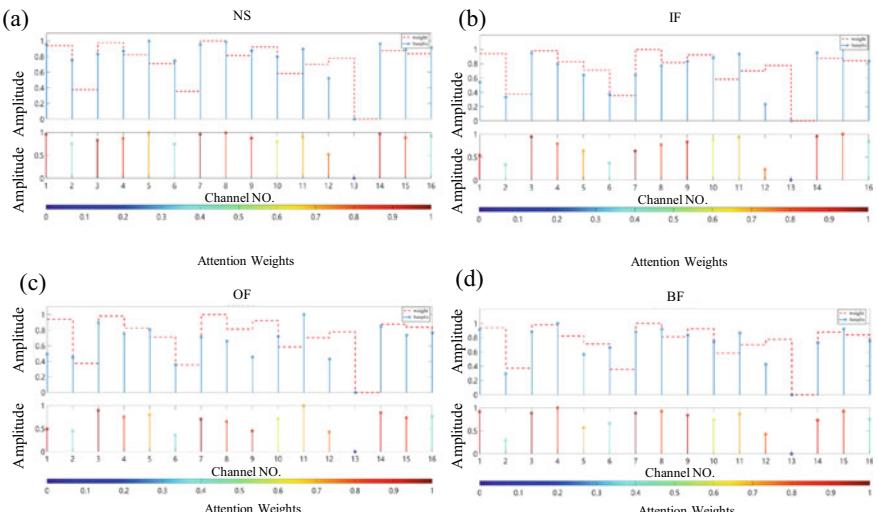


Fig. 12 Visualization of attention weights and sample entropy. **a** NS. **b** IF. **c** OF. **d** BF

5 Conclusion

To address the problem of limited labeled data and to reduce the influence of noise environment on model robustness, a multi-scale convolutional autoencoder with attention mechanism (MSCAE-AM) is proposed for fault diagnosis under noisy environments. The autoencoder improves the feature extraction ability of the model in noisy environments by combining noise reduction operations and embedding multi-scale convolutional layers and attention mechanisms. The effectiveness of the proposed method is verified through the case study, which demonstrates the superiority of the proposed method in industrial practice. Furthermore, the developed method can enhance the applicability of unsupervised learning techniques in robust fault diagnosis to facilitate the application of intelligent maintenance in real industrial scenarios, which will help to address the challenges of robust fault diagnosis with limited labeled data sets and different background noises.

In future work, few-shot learning will be considered for intelligent maintenance, including different degrees of data imbalance. Additionally, research will continue on more advanced lightweight models for deployment in industrial scenarios.

Acknowledgements This work was supported in part by the National Key Research and Development Program of China (No. 2020YFB1710002), in part by the National Research Council of Canada through the Canada—Germany 3+2 Joint Project “Digital Twin Platform for Infrastructure Asset Lifecycle Management” (Agreement No. INT-016-1).

References

1. Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., & Nandi, A. K. (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing.*, 138, 106587.
2. Wang, D., Peter, W. T., & Tsui, K. L. (2013). An enhanced Kurtogram method for fault diagnosis of rolling element bearings. *Mechanical Systems and Signal Processing.*, 35(1–2), 176–199.
3. Lei, Z., Chen, H., Wen, G., Feng, K., Liu, Z., Yan, R., et al. (2023). A Synchronous Holobalancing Method for Flexible Rotors Based on the Modified Initial Phase Vector. *Information Fusion.*, 90, 95–110.
4. Hou, B., Wang, D., Peng, Z., Tsui, K. -L. (2023). Adaptive fault components extraction by using an optimized weights spectrum based index for machinery fault diagnosis. *IEEE Transactions on Industrial Electronics.*
5. Lei, Z., Wen, G., Dong, S., Huang, X., Zhou, H., Zhang, Z., et al. (2020). An intelligent fault diagnosis method based on domain adaptation and its application for bearings under polytropic working conditions. *IEEE Transactions on Instrumentation and Measurement.*, 70, 1–14.
6. Xu, Y., Yan, X., Feng, K., Sheng, X., Sun, B., & Liu, Z. (2022). Attention-based multiscale denoising residual convolutional neural networks for fault diagnosis of rotating machinery. *Reliability Engineering and System Safety.*, 226, 108714.
7. Zhao, Z., Wu, J., Li, T., Sun, C., Yan, R., & Chen, X. (2021). Challenges and opportunities of AI-enabled monitoring, diagnosis and prognosis: A review. *Chinese Journal of Mechanical Engineering.*, 34(1), 1–29.
8. Zhang, W., Li, X., & Ding, Q. (2019). Deep residual learning-based fault diagnosis method for rotating machinery. *ISA Transactions.*, 95, 295–305.

9. Lei, Z., Zhang, P., Chen, Y., Feng, K., Wen, G., Liu, Z., et al. (2023). Prior knowledge-embedded meta-transfer learning for few-shot fault diagnosis under variable operating conditions. *Mechanical Systems and Signal Processing*, 200, 110491.
10. Jia, F., Lei, Y., Lu, N., & Xing, S. (2018). Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization. *Mechanical Systems and Signal Processing*, 110, 349–367.
11. Guo, X., Shen, C., & Chen, L. (2016). Deep fault recognizer: An integrated model to denoise and extract features for fault diagnosis in rotating machinery. *Applied Sciences*, 7(1), 41.
12. Chen, Z., & Li, Z. (2018). Fault diagnosis method of rotating machinery based on stacked denoising autoencoder. *Journal of Intelligent and Fuzzy Systems*, 34(6), 3443–3449.
13. Zhang, W., Peng, G., Li, C., Chen, Y., & Zhang, Z. (2017). A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors*, 17(2), 425.
14. Wang, H., Liu, Z., Peng, D., & Cheng, Z. (2022). Attention-guided joint learning CNN with noise robustness for bearing fault diagnosis and vibration signal denoising. *ISA Transactions*, 128, 470–484.
15. Xu, Y., Yan, X., Feng, K., Zhang, Y., Zhao, X., Sun, B., et al. (2023). Global contextual multiscale fusion networks for machine health state identification under noisy and imbalanced conditions. *Reliability Engineering and System Safety*, 231, 108972.
16. Li, S., Ji, J., Xu, Y., Sun, X., Feng, K., Sun, B., et al. (2023). IFD-MDCN: Multibranch denoising convolutional networks with improved flow direction strategy for intelligent fault diagnosis of rolling bearings under noisy conditions. *Reliability Engineering and System Safety*, 237, 109387.
17. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudny, A., Shuai, B., et al. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, 77, 354–377.
18. Wang, Y., Yao, H., & Zhao, S. (2016). Auto-encoder based dimensionality reduction. *Neurocomputing*, 184, 232–242.
19. Li, P., Chen, Z., Yang, L. T., Gao, J., Zhang, Q., & Deen, M. J. (2018). An improved stacked auto-encoder for network traffic flow classification. *IEEE Network*, 32(6), 22–27.
20. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
21. Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. -A. (Eds.). (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on machine learning*.
22. Van der Maaten, L., Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11).
23. Pincus, S. M., & Huang, W.-M. (1992). Approximate entropy: Statistical properties and applications. *Communications in Statistics-Theory and Methods*, 21(11), 3061–3077.