



A Survey on Generative Adversarial Networks: Variants, Applications, and Training

ABDUL JABBAR, XI LI, and BOURAHLA OMAR, College of Computer Science, Zhejiang University, Hangzhou, China

The Generative Models have gained considerable attention in unsupervised learning via a new and practical framework called Generative Adversarial Networks (GAN) due to their outstanding data generation capability. Many GAN models have been proposed, and several practical applications have emerged in various domains of computer vision and machine learning. Despite GANs excellent success, there are still obstacles to stable training. The problems are Nash equilibrium, internal covariate shift, mode collapse, vanishing gradient, and lack of proper evaluation metrics. Therefore, stable training is a crucial issue in different applications for the success of GANs. Herein, we survey several training solutions proposed by different researchers to stabilize GAN training. We discuss (I) the original GAN model and its modified versions, (II) a detailed analysis of various GAN applications in different domains, and (III) a detailed study about the various GAN training obstacles as well as training solutions. Finally, we reveal several issues as well as research outlines to the topic.

CCS Concepts: • Computing methodologies → Artificial intelligence; Computer vision; Computer vision problems; Reconstruction;

Additional Key Words and Phrases: Generative Adversarial Networks (GANs), architectural-variants, applications, stabilize training

ACM Reference format:

Abdul Jabbar, Xi Li, and Bourahla Omar. 2021. A Survey on Generative Adversarial Networks: Variants, Applications, and Training. *ACM Comput. Surv.* 54, 8, Article 157 (October 2021), 49 pages.

<https://doi.org/10.1145/3463475>

1 INTRODUCTION

Most of the techniques used in **Artificial Intelligence (AI)** are supervised machine learning, whereas unsupervised learning is still a relatively unsolved research area. Recently, generative modeling, considerable with deep learning techniques, opened a new hope in the field of unsupervised learning, and **Generative Adversarial Networks (GAN)** is one of them. GAN is an example of generative models presented by Goodfellow et al. [1]. GAN is the most common

This work is supported in part by key scientific technological innovation research project by Ministry of Education, Zhejiang Provincial Natural Science Foundation of China under Grant LR19F02000, National Natural Science Foundation of China under Grant U20A2022 and National Key Research and Development Program of China under Grant 2020AAA0107400.

Authors' addresses: A. Jabbar, X. Li (corresponding author), and B. Omar, College of Computer Science and Technology, Yuquan Campus, Zhejiang University, Hangzhou, Zhejiang, China, 310007; emails: {jabbar, xilizju, obourahla}@zju.edu.cn. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.

0360-0300/2021/10-ART157 \$15.00

<https://doi.org/10.1145/3463475>

learning model in both semi-supervised and unsupervised learning. Theoretically, GAN takes a supervised learning approach to do unsupervised learning by generating fake or synthetic-looking data. The soul of GAN can be summated as training of two networks simultaneously called the generator network (G) and the discriminator network (D). D is a classifier that is used to differentiate the real and fake data as genuinely as possible. In contrast, G confuses D by generating real data. These two networks section themselves, and eventually G produces realistic data, and D gets better at predicting the fake ones.

Practically, GANs have introduced many applications such as handwritten font generation [28–34], anime characters generation [35, 39], image blending [40–43], image inpainting [44, 45], face aging [49–53], text-to-image synthesis [54–58], human pose synthesis [59, 60], stenographic applications [61–63], image manipulation applications [64, 68], visual saliency prediction [69–71], object detection [72–75], three-dimensional (3D) image synthesis [76–78], medical applications [81–88], facial makeup transfer [89–91], facial landmark detection [92, 93], image super-resolution [94–97], texture synthesis [98–101], sketch synthesis [102, 104–106], image-to-image translation [27, 107–115], face frontal view generation [116–120], language and speech synthesis [144, 145], music generation [146, 148, 149], video applications [162–165], and climate and earth science applications [170, 171].

Although GANs have achieved some incredible results by producing stunningly realistic samples, it is still difficult to train the GAN through better stability. Stable GAN training is a crucial problem, because both G and D need to optimize through alternating gradient descent or simultaneous gradient descent. GAN architecture suffers due to several shortcomings, such as the Nash equilibrium [206], internal covariate shift [207], mode collapse [208, 209], vanishing gradient [208], and lack of proper evaluation metrics [210]. Thus, several training solutions such as feature matching [211], unrolled GAN [212], mini-batch discrimination [211], historical averaging [211], two timescale update rule [213], hybrid model [214, 215], self-attention GAN [216], relativistic GAN [220], one-side label smoothing [221], sampling GAN [222], proper optimizer [226], normalization techniques [221, 243–248], spectral normalization [235], add noise to inputs [227, 228], training with labels [15], alternative loss functions [229], gradient penalty [21], representative features based GAN [230], and cycle-consistency loss [108] have been introduced for many years to stabilize GAN training. We try to present a GAN survey from its variants, applications, and training perspective in this attempt.

1.1 Structure of This Article

The structure of this article is as follows. First, a concise introduction of the GAN and GAN-variants as shown in Table 1 is given. In addition, an extensive comparative analysis of GAN-variants is presented. Second, numerous expansions of the GAN practiced in different areas, as demonstrated in Table 3 are given. Third, we briefly survey several problems related to the GAN stable training and their solutions to increase GAN's training stability. Finally, we conclude this article. In particular, we address several new issues and potential future research directions on the topic.

1.2 Relevance to Other Surveys and Significance

Recently, several surveys (e.g., Gui et al. 2020 [2]; Wang et al. 2019 [3]; PAN et al. 2019 [4]; Hong et al. 2019 [5]; Hitawala 2018 [6]; Creswell 2018 [7]; Vuppuluri et al. 2017 [8]) have explored the GANs algorithms and their potential applications, as Table 2 shows the list of most related survey papers of GAN. Among these surveys, Gui et al. (2020) [2] survey provide a review of different GANs methods from theory, algorithms, and application perspectives. The study of Wang et al. (2019) [3] presents a descriptive, restricted GAN overview, focusing exclusively on the GANs

Table 1. A Summary of GAN-variants Discussed in This Article

#	Subject	Pub.	Year
1.	Conditional Generative Adversarial Networks (C-GAN) [15]	arXiv	2014
2.	Deep Convolution Generative Adversarial Networks (DC-GAN) [16]	CoRR	2015
3.	Laplacian Generative Adversarial Networks (Lap-GAN) [17]	NIPS	2015
4.	Information Maximizing Generative Adversarial Networks (Info-GAN) [18]	NIPS	2016
5.	Energy-Based Generative Adversarial Networks (EB-GAN) [19]	ICLR	2016
6.	Wasserstein Generative Adversarial Networks (W-GAN) [20]	ICML	2017
7.	Boundary Equilibrium Generative Adversarial Networks (BE-GAN) [22]	CSLG	2017
8.	Progressive-Growing Generative Adversarial Networks (PG-GAN) [23]	ICLR	2018
9.	Big Generative Adversarial Networks (Big-GAN) [24]	ICLR	2019
10.	Style Generator Architecture for Generative Adversarial Networks (Style GAN) [25]	IEEE	2019

Table 2. A Summary of Related Literature Surveys

#	Subject	Pub.	Year
1.	A Review on Generative Adversarial Nets: Algorithms, Theory, and Applications [2]	arXiv	2020
2.	Generative Adversarial Networks: A Survey and Taxonomy [3]	arXiv	2019
3.	Recent Progress on Generative Adversarial Networks (GAN): A Survey [4]	IEEE	2019
4.	How Generative Adversarial Nets and its Variants Work. An Overview of GAN [5]	ACM	2019
5.	Comparative Study on Generative Adversarial Networks [6]	CoRR	2018
6.	Generative Adversarial Networks: An Overview [7]	IEEE	2018
7.	Survey on Generative Adversarial Networks [8]	IJERCS	2017

architectural-variants and the loss function-variants only relevant to computer vision. The survey of PAN et al. (2019) [4] summarizes the GAN’s background, its types, and its applications in several fields before 2019. The study of Hong et al. (2019) [5] summarizes the current GAN landscape but leaves out several important aspects crucial for stabilizing GAN training. The survey of Hitawala (2018) [6] provides a comparative analysis of GAN and GAN-variants. The survey of Creswell (2018) [7] provides a summary of GAN for the community of signal processing. The survey of Vuppuluri et al. (2017) [8] attempts to compare and comprehend different GAN applications. These surveys summarize the whole field, outlining key GAN concepts, methods, and applications; they do so without an explanation of the instability dilemma and GAN training techniques.

The main difference of this article from previous work is summarized as follows.

- Our article provides a more comprehensive review of GAN training obstacles, basic reasoning behind training obstacles, and an in-depth revision of training techniques with their basic properties.
- Our article conducts experiments to quantitatively evaluate the effects on the training stability of the several training techniques. It performs a comparative analysis to show which training technique has the most significant effect on GAN’s training process.
- Our article also aims to present a comparative analysis across GAN training techniques in terms of their pros and cons to better understand their strengths and limitations, because most of the training techniques are problem based, i.e., each stabilization technique is appropriate for a particular situation/problem.

Table 3. A Summary of GAN Practical Applications Reviewed in This Article

Dom.	Subject with applied model names
Image	Handwritten font generation: Zi2zi [28], DenseNet-CycleGAN [29], LS-CGAN [30], GlyphGAN [31], MCMS-CGAN [32], HW-GAN [33], Ss-GAN [34] Anime characters generation: Anime character with GAN [35], PS-GAN [39] Image blending: GP-GAN [40], GCC-GAN [43] Image inpainting: Ex-GAN [44], CA-GAN [45], PG-GAN [46] Face aging: Age-CGAN [49], CAAE [50], IP-CGAN [51], Wavelet-GANs [52, 53] Text-to-image: AttnGAN [54], StackGAN [55], ACGAN [56], TACGAN [57], SISGAN [58] Human pose synthesis: PG ² [59], Deformable-GAN [60] Stenographic applications: SGAN [61], SS-GAN [62], Stegano-GAN [63] Image manipulation: IGAN [64], TA-GAN [65], IAN [66], Att-GAN [67], D-GAN [68] Visual saliency prediction: Sal-GAN [69], SalCapsule-CGAN [70], DSAL-GAN [71] Object detection: Se-GAN [72], P-GAN [73], SOD-MTGAN [74], GAN-DO [75] 3D image synthesis: 3D-GAN [76], Pr-GAN [77], 3D-CGAN [78] Medical applications: SegAN [81], Med-GAN [82], Chem-GAN [83], FB-GAN [84], GAN-based brain activity [85], Dental Restoration [86], Dr-GAN [87], MedGAN [88] Facial makeup transfer: B-GAN [89], PairedCycle-GAN [90], DM-TGAN [91] Face landmark detection: SAN [92], Expose-GAN [93] Image super-resolution: SR-GAN [94], ESR-GAN [95], SR-DGAN [96], T-GAN [97] Texture synthesis: M-GAN [98], S-GAN [100], PS-GAN [101] Sketch synthesis: T-GAN [102], Autopainter [104], Sketchy-GAN [105], CA-GAN [106] Image-to-image translation: pix2pix [27], PAN [107], Cycle-GAN [108], Disco-GAN [109], Dual-GAN [110], StarGAN [111], UNIT [112], MUNIT [113], DRIT [114], DRIT++ [115] Face frontal view generation: DRGAN [116], TPGAN [117], FFGAN [118], FTGAN [120]
Audio	Speech and audio synthesis: Rank-GAN [144], VAW-GAN [145] Music generation: C-RNN-GAN [146], Seq-GAN [148], OR-GAN [149]
Video	Video applications: VGAN [162], MoCoGAN [163], DRNET [164], DVD-GAN [165]
Climate & earth science	State-Parameter Identification GAN [170], 4D-Seismic-Inversion GAN [171]

—Our article tries to summarize, compare, and examine the recent best generative models with specific emphasis on the GAN landscape covering all its aspects, from its architectural and loss function variants to its various domains applications perspective.

This article makes the following contributions:

- We describe and compare the best generative models with an explicit focus on GAN background from variants, training, and application not only related to computer vision and image processing communities but also in other associated areas of research such as natural language processing, medical field, earth, and climate science, and so on. Also, an in-depth study and a brief comparison of GAN-variants in-terms of their merits and demerits are provided.
- We critically review several training obstacles and discuss several training techniques with their properties in-details.
- We conduct experiments to assess the GAN training stability of different stabilization techniques and perform a comparative analysis among training techniques.
- We discuss several new issues to highlight various open problems for the progress of the GANs.

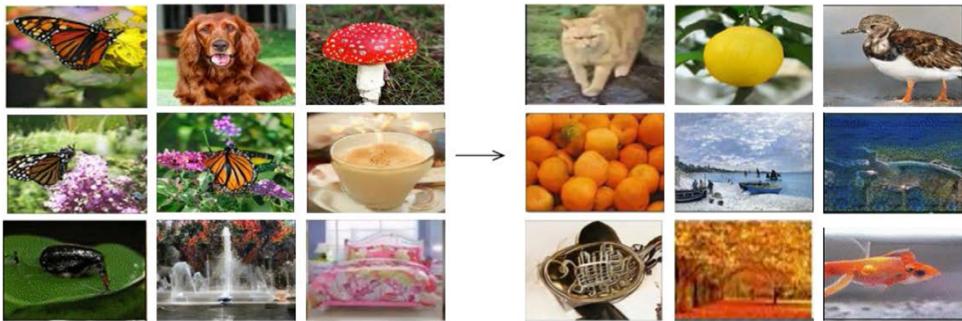


Fig. 1. Some good generative models can train on data examples and then generate more sample data. The left side shows the training, and the right-side shows the generated samples. Image regenerated from Reference [26].

2 BACKGROUND

Generative models (GM) are rapidly advancing the research area of **computer vision (CV)**. Generative models are the classical models for unsupervised learning. The training data $\sim p_{data}(x)$ from an unknown data-generating distribution generates new samples data $\sim p_{model}(x)$ from the same distribution. The end goal of any GM is to draw similar data samples ($p_{model}(x)$) from the leaned real data distribution ($p_{data}(x)$) (see Figure 1), which is also best explained with the help of the following training objective:

Generated data ($p_{model}(x)$) want to be similar to training data ($p_{data}(x)$).

Why the generative model?

- Generation of realistic samples.
- Handle the missing data well.
- Training of GM enables the inference of latent representations can be useful as a general feature.
- Address the density estimation problem in unsupervised learning.
- It solves the problem of generating new data for training without human supervision and interventions. Generative models are essential in the perspective of modern AI.

2.1 Autoencoder

Autoencoder (AE) is the popular type of generative model that takes high-dimensionality data and compresses it into a small representation with simple neural networks without massive data loss [9]. Any AE contains two types of networks: encoder and decoder. The encoder is a bunch of layers that takes the input data and compresses it down to a small demonstration, which has fewer dimensions. This low or compressed demonstration of input data is called a bottleneck. The decoder takes that bottleneck and tries to reconstruct the input data (see Figure 2). The AE calculates the reconstruction loss through per-pixel differences between encoder input and decoder output. The objective function of AE is mathematically defined as

$$L_{AE} = \frac{1}{n} \sum_{i=1}^n [x - f_{(\theta)}(g_{(\phi)}(x))]^2, \quad (1)$$

where $g_{(\phi)}$ represents the encoder network, $f_{(\theta)}$ represents the decoder network, x represents the input data, Φ , and θ represents the network parameters. A simple Euclidean distance calculates the reconstruction loss: $\| \text{input data} - \text{reconstructed data} \|^2$, i.e., pixel-by-pixel comparison.

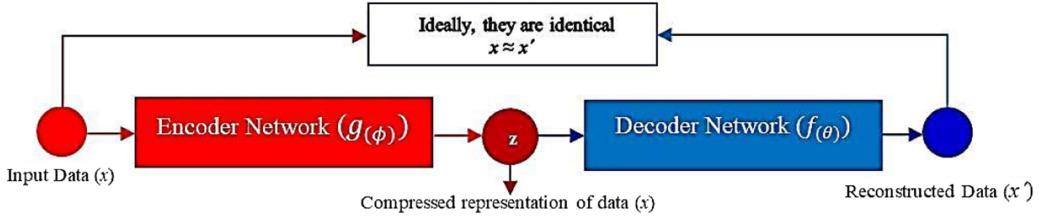


Fig. 2. The architecture of AE comprises two deep networks, encoder ($g_{(\phi)}$) compresses the input data (x) through system define parameters and decoder ($f_{(\theta)}$) decompress the compressed data (z).

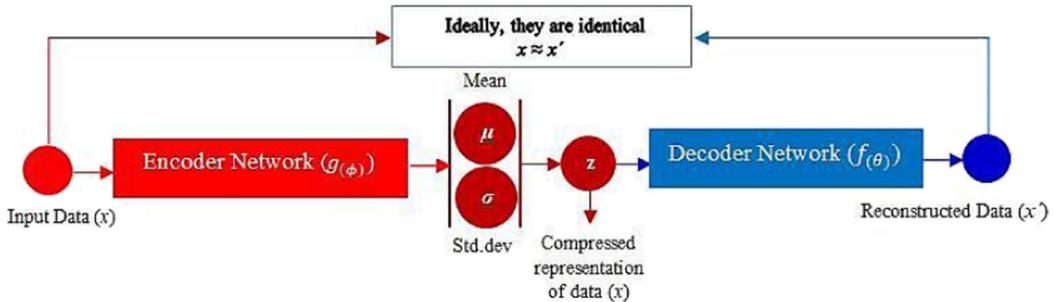


Fig. 3. The architecture of VAE comprises a probabilistic encoder ($q_{\theta}(z|x)$) and a probabilistic decoder ($p_{\theta}(z|x)$).

2.2 Variational Autoencoder

Variational Autoencoder (VAE) is another widely used likelihood-based generative model. It includes a probabilistic encoder network (parameterized by ϕ), a probabilistic decoder network, or a generative network (parameterized by θ) and loss functions [10]. The probabilistic encoder ($q_{\theta}(z|x)$) (also called latent variable generative model) embeds a data sample (x) into discrete latent variables is denoted by (z) and probabilistic decoder network ($p_{\theta}(z|x)$) reconstructs the input sample based on the discrete latent vector (z) without massive input data loss (see Figure 3). The cost function of the VAE is defined as

$$L_{VAE} = \mathbb{E}_{q_{\theta}(z|x)} [\log p_{\theta}(xz)] - D_{KL}[(q_{\theta}(zx) || p_{\theta}(z)], \quad (2)$$

where x represents the real data distribution and θ and θ represent the parameterize distribution for VAE probabilistic encoder-decoder.

2.3 Generative Adversarial Networks

GAN is a robust network used for unsupervised machine learning to build a min-max game between two-players, i.e., setting up both the player (networks) with their different objectives. The first player (generator) tries to fool the second player (discriminator) by producing very natural-looking real-world images from a random latent vector (z), and the second player (discriminator) gets better at distinguishing the fake and the real data (see Figure 4). Both networks try to optimize themselves in the best way to accomplish the individual objectives, because both have their objective functions, i.e., the discriminator wants is to maximize its cost value, and the generator wants to minimize its cost value. The GAN training objective is defined as follows:

$$V_{GAN}(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log(D(x))] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] , \quad (3)$$

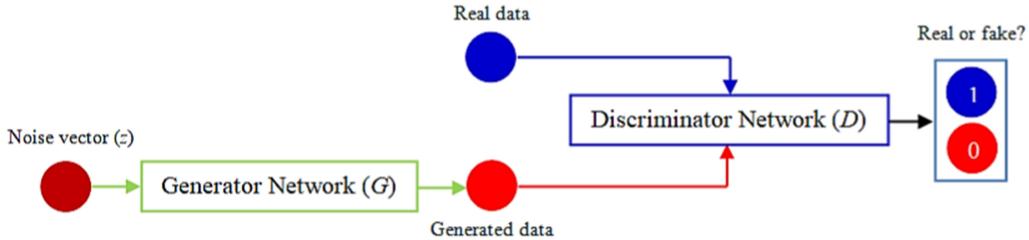


Fig. 4. The architecture of the GAN, where the objective of the discriminator is to maximize its cost value, i.e., $\log(D(x))$ and the generator is to minimize its cost value, i.e., $\log(1-D(G(z)))$.

where Equation (3) shows that there are two loss functions $-\log(D(x))$ for the discriminator and $\log(1-D(G(z)))$ for the generator and two optimizers for the G and D , since they are two different networks.

2.3.1 Analysis of GAN Merits and Demerits. GAN is arguably the first and arguably the best AI model to achieve very realistic and consistent results. GAN has impressive results but shortcomings as well. The following sections aim to present an analysis of the merits and demerits of the GAN.

Merits: GAN has great significance to the development of generative models. As a powerful class of generative models, GAN has the following merits. (1) It can generate very sharp and real-life samples of various kinds, including high-quality images, videos, and audio indistinguishable from real data (state-of-the-art samples). (2) The adversarial networks of GAN manage to learn its cost function, i.e., its own complicated rules of what is right and wrong—overcoming the need to carefully design and build a cost function. (3) GAN enables machine learning to work with a single input and multimodal outputs. (4) GAN performs unsupervised learning tasks reasonably well as compared to other generative models. (5) It has fast inference (i.e., when you have some trained model, you can generate objects quickly). (6) The training process of GAN is not limited to only the available training dataset.

Demerits: Despite the great successes GAN has achieved, some shortcomings still need to be improved. (1) GAN does not work with any explicit density function; instead, it learns to generate from training distribution through a two-player game. (2) GAN's model is more challenging to optimize due to unstable training dynamics. (3) Inverting is not straightforward (GAN is not trained to be invertible). (4) It cannot generate better text or speech data, because GAN defines continuous real value data. (5) Despite the GAN grand success, it lacks intrinsic evaluation metrics. (6) GAN has a higher computational cost in the sense that it requires high-powered GPUs to produce good results.

2.4 Glow

Glow is a flow-based generative model [11] that shows improved performance in terms of log-likelihood compared to earlier popular generative models, such as GAN [1] and VAE [10]. The key idea behind this multi-scale architecture is a series of flow steps such as (I) Actnorm Layer-affine transformation of the activations functions through the bias-parameter and channel-scale for normalization functions, (II) Invertible 1×1 convolution-fixed permutation changed with learned 1×1 convolution whose weight matrix initialized as a random rotation matrix, and (III) Affine Coupling Layers-coupling multiple layers for numerical stability. The critical properties of the flow-based generative models are as follows:

- Precise latent variable presumption and log-likelihood estimation:
 - VAE infer only approximately the value of the latent.
 - GAN has no encoder at all to infer the latent.
 - Glow directly infers the latent and optimizes the exact log-likelihood of the data.
- Efficient inference and efficient synthesis:
 - Autoregressive models, such as the PixelCNN [14], are inefficient on parallel hardware.
 - Glow is efficient to parallelize.
- Valuable latent space for downstream tasks:
 - Autoregressive models have unknown marginal distributions.
 - In GAN, data points cannot usually directly represented in a latent space.
 - Glow and VAE infer meaningful latent.
- Significant potential for memory savings:
 - Computing gradients require a constant amount of memory.

2.5 Vector Quantized Variational AutoEncoder-2

Vector Quantized VAE-2 (VQ-VAE-2) [12] is an improved version of **Vector Quantized VAE (VQ-VAE)** [13] that generates a large size image. The key idea behind this simple feed-forward encoder and decoder architecture is using discrete latent space representation means that instead of thinking in pixels, it thinks more in terms of features that commonly appear in natural photos, which also makes the generation of these images up to 30 times quicker, and super-useful in case of larger images. This technique rapidly generates new diverse images with a size of approximately $1,000 \times 1,000$ pixels. VQ-VAE-2 is a two-stage training process. Stage-I of VQ-VAE-2 first encodes the input image into two different latent maps: local information and another for the input image's global details. Stage-II of VQ-VAE-2 imposes an Autoregressive model [14] on both latent space levels to compress the compressed image during stage-I. Then the simple feed-forward decoder reconstructs the original size image by taking all levels of latent input maps. VQ-VAE based generative models are robust against mode collapse and lack of diversity shortcomings compared to rivalry GAN.

2.6 Difference/Relationship between GAN and VAE/Glow/VQ-VAE-2

Recent research suggests that different popular deep generative models such as GAN [1], VAE [10], Glow [11], and VAE-VQ-2 [12] show excellent performance in the generation of natural-looking realistic samples. However, these popular deep generative models have various performance features and tradeoffs. We used the sample evaluation metrics, sample diversity, sample interpolation, and computational cost parameters to explain the significant difference/relationship between GANs and VAE/Glow/VQ-VAE-2. Table 4 presents a summary of significant difference/relationship between GANs and VAE/Glow/VQ-VAE-2 discussed in the survey.

- **Sample evaluation metrics:** One of the principal motivations to use VAE, Glow, and VAE-VQ-2 is to optimize the **Negative-log Likelihood (NLL)** objective function for the model's evaluation. The NLL of the training and test data gives an objective measure for a perfect score due to its generalization performance to hidden training data. However, the GAN type's generative models do not have any standard defined function for assessment. Most of the proposed best GANs-derived models introduced their new evaluation metrics. Thus, there are no robust and standard consensus parameters for a fair comparison of GAN-derived models, which hurts the GANs' performance. Mostly sample-quality and sample-diversity are evaluated via visual fidelity of the samples generated by GAN methods.

Table 4. A Summary of the Difference/Relationship between GAN and VAE/Glow/VQ-VAE-2

Parameter	Difference/Relationship
Sample evaluation metrics	Deep generative models such as VAE, Glow, and VQ-VAE-2 optimized the NLL objective function for model evaluation, whereas for GAN, there is no standard defined function for assessment.
Sample diversity	A deep generative model such as GAN fails to capture the diversity in the training data compared to VAE, Glow, and VQ-VAE-2 has the potential to capture the diversity.
Sample interpolation	All discussed generative models allow interpolation between the generated samples except the VQ-VAE-2 generative model.
Computational cost (Sampling time/speed)	Deep generative models such as GAN and Glow have smaller computational cost due to faster sampling times/speed. In contrast, VAE and VQ-VAE-2 have greater computational cost due to slower sampling times/speed.

- **Sample diversity:** Many authors have pointed out that GAN can generate extraordinarily, realistic, high-quality samples at high resolution; however, GAN may fail to capture all the real data's diversity. The learned model (GAN) covers only one or a few goal distribution modes instead of covering all possible training modes. This leads the model to a state usually referred to as mode collapse, where the model fails to capture the proper data distribution. Thus, GAN-generated samples are inferior in diversity, because they do not entirely contain the real data distribution diversity. On the contrary, VAE, Glow, and VAE-VQ-2 can capture the data's diversity, because the probability that the model allocates to all samples in the training data is maximized. In-addition, VAE, Glow, and VAE-VQ-2 shows better performance against training instability issue of mode collapse and lack of diversity, because their generated samples entirely capture all modes of true data distribution.
- **Sample interpolations:** Many generative models, such as VAE, GAN, and Glow, allow interpolation between generated samples. This is performed by first interpolating, linearly or otherwise, between the input images' latent vectors and then generates the new image from these interpolated latent vectors. However, for the VQ-VAE-2 generative model, there exists no way, i.e., linearly or otherwise, to interpolate between a set of generated samples.
- **Computational cost:** Sample generation cost is another important parameter used by different researchers to show the relationship/difference between different generative models. The recent literature review suggests that the model's computational cost depends on how faster the sampling times/speed. Some deep generative models such as GANs and Glow only take the single forward pass to compute all the sample dimensions in training data. Some deep generative models, such as VAE and VQ-VAE-2, take many forward passes to compute all the sample dimensions in training data. Thus, GANs and Glow models have considerably faster sampling times over VAE and VQ-VAE-2. This faster sampling time/speed reduces the computational cost of the model.

3 GAN VARIANTS

Due to the popularity and success of GAN, many adjustments and variations over the original GAN model have been proposed to improve training. Supplementary Material discuss some of the modified versions of GANs along-with GANs-variants analysis in-terms of their evaluation metrics, because prior knowledge about the baseline model's evaluation metrics may be crucial before applying it successfully for a practical application.

4 APPLICATIONS

GAN is an exceptionally excellent generative model in generating photo-realistic samples after the models have trained on some data. These benefits lead GAN to be functional in different computer vision and image processing fields and other related research areas. In this section, we discuss various GAN applications in image, audio, video, and climate and earth science domains. Supplementary Material provides a much needed in-depth analysis of GAN applications merits and demerits discussed in this survey.

4.1 Image Domain

4.1.1 Handwritten Font Generation. Automatic character generation is a difficult task even becomes more challenging. The Chinese language has plenty of logographic characters in distinction to the world's phonological languages, such as English or French. Zi2zi framework is a CGAN-based Chinese character generation model [28] that is directly derived and extended from the popular **pix-to-pix translation (pix2pix)** [27] model. Zi2zi uses the same architecture as pix2pix with an additional category embedding network for multiple Chinese fonts. Zi2zi uses Chinese character paired training data of source and target fonts. Zi2zi uses paired training data for handwritten Chinese character generation, and the collection of paired training data is a difficult task. Therefore, it is not appropriate to use paired Chinese characters for the handwritten Chinese generation instead of unpaired Chinese characters for the same problem.

The DenseNet-CycleGAN [29] is a **Cycle-consistent adversarial networks- (Cycle-GAN)** [108] based model proposed for Chinese handwritten character generation with unpaired training data of source and target fonts. The DenseNet-CycleGAN uses to study a mapping from a presented printed font to a modified handwritten style where the source style information is first encoded via the encoder network to a lower-dimensional space. The features extracted move through a transfer unit considered the features extracted in the target style. Finally, the transfer unit output through the decoder network in the form of handwritten Chinese characters. The objective function uses two losses, such as adversarial loss that enforce the realistic Chinese characters generation and the cycle-consistency loss that enforce an image to be transformed back to itself to generate high-quality Chinese handwritten characters with a promising rate of success. The authors also proposed two metrics to measure their generated Chinese characters' quality: content accuracy and style discrepancy. For content accuracy, a classifier model is trained on the reference data, and then it is used to identify the characters generated by their generator. For style discrepancy, a root-mean-square calculates the dissimilarity in styles between the real and generated characters.

Least Squares CGAN (LS-CGAN) [30] proposed a stroke-based Chinese font generation model by combining a range of Chinese characters. The LS-CGAN generates a new Chinese character font style by fusing the existing Chinese character's font styles. LS-CGAN's training phase consists of two parts: offline LS-CGAN training, which includes stroke extraction, condition vector generation, and LS-CGAN training. The LS-CGAN's on-line font generation process involves the strokes extraction, generation of new fonts, and new font strokes. Through adjusting the weight factor, two qualified fused styles can be interpolated. The GlyphGAN [31] is a GAN-based Chinese font generation model that takes two input vectors as a class character vector associated with the character class information and a style-vector associated with the style information. The GlyphGAN model generates diverse font characters by keeping the same style across all aspects. The GlyphGAN generator produces synthetic fonts similar to those manually designed ones by taking the input vector consisting of the style vector and the character type class vector. The GlyphGAN discriminator classifies the generator-generated fonts from real fonts via Wasserstein distance calculation between the synthesized and real data distributions. The loss function of GlyphGAN uses



Fig. 5. Examples of DRA-GAN-based generated anime characters' faces. Image from Reference [35].

L1 distance loss between the fake and real font images. The GlyphGAN model can produce an endless range of fonts with the character and style separately proscribed.

Furthermore, GAN has also been utilized in many other character generation applications such as a **Multi-Scale Multi-Class CGAN (MCMS-CGAN)** [32] model that generates multi-class realistic and natural-looking Chinese handwritten characters. The generator of MCMS-CGAN generates realistic Chinese handwritten characters under multi-class conditional information. The MCMS-CGAN discriminator discriminates against the generator-generated Chinese handwritten characters from real Chinese handwritten characters. **Handwritten GAN (HW-GAN)** [33] is another model proposed for synthesizing handwritten stroke data. As standard GANs architectures, the HW-GAN generator generates realistic handwritten text, and a discriminator discriminates the realistic handwritten text from generator-generated forgery ones. A **Semi-Supervised GAN (Ss-GAN)** [34] proposed for the generation of Bangla handwritten characters. The S-GAN generator takes the random noise vector as input and generates the Bangla handwritten image. The S-GAN discriminator discriminates the generator generated fabricated Bangla handwritten sample from a real handwritten sample of the real Bangla handwritten digits dataset.

4.1.2 Anime Character Generation. The creation of animated characters in computer games is expensive. Cost-effective anime-character generation is possible and requires a reduced number of artistic skills with GAN. The automatic anime character generation with GAN [35] generates artificial anime-style characters at high-quality with an encouraging rate of success by introducing a new gradient penalty method to improve the learning stability of GAN, called **Deep Regret Analytic GAN (DRA-GAN)** [36]. They used DRA-GAN as the basis of their GAN model to generate pleasing fine anime character faces images with small distortion. The architecture of automatic anime character with GAN is composed of a generator, which is an adaptation from SRResNet [94], has 16 ResBlocks with three sub-pixel CNN [37], to up-scale the feature map, and a discriminator network has 10 ResBlocks with no batch normalization layer [207] to bring the relationships inside the mini-batch, is not required for gradient norm calculation. The experiments on the Japanese anime characters dataset [38] have shown that DRA-GAN-based models can presumably generate real, diverse facial images. It has the least computational cost compared to other GAN-variants, because DRA-GAN-based models successfully minimize the effects of mode collapse during training. Some experimental results of this DRA-GAN-based automatic anime character generation model (see Figure 5).

Following the recently proposed idea of **progressive-growing GAN (PG-GAN)** [23], the **Progressive Structure-Conditional Generative Adversarial Networks (PS-CGAN)** [39] model was proposed, which generates smooth, high-resolution (e.g., $1,024 \times 1,024$), and full-body anime characters. The PS-CGAN can create controllable animations for various styles with particular target poses by employing latent variables and structural information as applied conditions. The generator of PS-CGAN generates an anime character from a latent variable via applied conditions

such as facial landmarks and poses key-points. The PS-CGAN discriminator distinguishes the generated anime character from real anime character based on the applied conditions. The PS-GAN uses the same loss function as Reference [23]. The practical implementations of PS-GAN show that it can successfully generate better quality images of anime characters by following the progressive, growing mechanism.

4.1.3 Image Blending. The blending of images together shows a dominant performance in several computer vision tasks, such as modification of visual communication or automatic photo editing. **Gaussian Poisson GAN (GP-GAN)** [40] proposed an image blending method for high-resolution input images. The GP-GAN framework combines the C-GAN [15] model with the strengths of traditional gradient-based methods of image blending [41]. The GP-GAN is the first framework that used GAN in the blending of images that have very high-resolution. They suggested the Gaussian Poisson equation [42] to produce a blending image at a higher resolution. GAN generates the preliminary blended image with low-resolution, and the result is the optimization of the Gaussian-Poisson equation. The GP-GAN fuses the information by optimizing the Gaussian Poisson equation to produce a well-blended high-resolution image while preserving the higher resolution information.

Recently, **Geometrically and Color Consistent GAN (GCC-GAN)** [43] combines the foreground and the background of two images seamlessly from different sources. The GCC-GAN framework comprised of four sub-networks: (i) **Transformation Network (TN)** that generate a real-looking composite image by considering the geometric and color consistency, (ii) Refinement Network that sharpens the boundary-edges of TN generated compound image, (iii) Discriminator Network that distinguishes the newly generated composite image from the real image and (iv) Segmentation Network that learns to split the foreground and the background in the generated composite image. GCC-GAN's loss function includes an adversarial loss, a geometric loss, an appearance loss, and an adversarial segmentation loss. The GCC-GAN model can automatically generate high-quality and realistic-looking composite images than the previous best GAN-derived methods.

4.1.4 Image Inpainting. Refilling the absent pixels of an image is known as image inpainting. It is an advanced reconstruction technique in a photo and video editing applications. **Exemplar GAN (Ex-GAN)** [44] uses the exemplar image information as a reference image for eye-closing to eye-opening tasks and produces a high-quality result. They assumed this extra conditional information as available at an inference of time and corresponded directly to specific characteristics of the item of importance. The generator input is an RGB image that removes the parts to be painted, stacked with a binary mask showing the regions to be filled-in. The discriminator manages the complete face and the image zoomed-in portion that includes the eyes. Ex-GAN uses the same objective function as vanilla GAN, but both the G and D can take an example image as Ex-GAN input.

However, the Contextual attention model [45] undoubtedly increases the inpainting technique's efficiency by learning feature representations to match and attend the related background patches. The proposed model consists of a two-stage training process. The primary step is to extract the absent content roughly with dilated convolution after training on reconstruction loss. The secondary step is to combine the attention to context. The central concept of context attention is to use the uniqueness of an established patch for patch generation as convolution filters. The architecture of the two networks that generate samples is identical to UNET like architecture. The whole contextual attention model is trained with **Wasserstein Generative Adversarial Networks (W-GAN)** [20] loss and pixelwise L1 distance loss to improve the results.

However, current network solutions still induce undesired artifacts and noise to the repaired regions. The PG-GAN [46] proposed an idea by combining the global GAN [47] with a patchwise GAN approach [48] to design a new discriminator network. The PG-GAN generator input is a corrupted image and the repaired image as output. The PG-GAN discriminator aims to distinguish the repaired and real images. The PG-GAN's objective function uses losses, such as reconstruction loss, that enforce the high-quality image reconstruction and adversarial loss that enforces the realistic image generation. The PG-GAN has good results than the previous methods by aggregating local and global information.

4.1.5 Face Aging. Face aging anticipates how a person looks in the future, while age regression analyzes how a person looks in the past subject to his present face and other relevant information. The Age-CGAN [49] model was proposed for face aging of human faces to anticipate future looks and looks. The Age-CGAN not only generates facial aging of the target underneath the age condition but can also protect the facial identity of the target image. The Age-CGAN architecture consists of a conditional GAN and an encoder-decoder arrangement. The encoder takes an input face image, maps it to a latent vector. Then the CGAN of the Age-CGAN takes the latent vector with a target age condition, maps to a new face image while maintaining the target image identity. The objective function of the Age-CGAN uses an identity loss that keeps the unique information of the generated image as real as possible, a classification loss that enforces the generator to generate the facial image of the accurate age group, and an adversarial loss that enforces the realistic image generation.

Similarly, the **Conditional Adversarial Autoencoder (CAAE)** [50] model was proposed for age regression and progression of human faces. The CAAE architecture contains an encoder that first maps an input face image to a latent vector, a generator that generates a new face image while maintaining the target image identity by taking the latent vector and the target age condition as inputs, and two discriminators, where the first discriminator enforces the latent vector to be consistently distributed. The second discriminator enforces the generator-generated results to be more photo-realistic face images in a target age condition. The objective function of the CAAE is composed of an L2 distance loss that enforces the generator generated and original input face images to be similar to a newly generated face image under the applied age condition, two adversarial losses where the first adversarial loss imposes a uniform distribution, and the second adversarial loss imposes the realistic face image generation task and a total-variation loss that eliminates the ghosting things from the generated face image. The CAAE generates photo-realistic face images and offers more robustness against pose variation, facial expression, and occlusion problems.

To preserve more facial identity information, **Identity-Preserved CGAN (IP-CGAN)** [51] model was proposed for age synthesis. In IP-CGAN, the CGAN model is used as a photo-realistic face image generator under the given age condition, an identity-preserved model is used to preserve the facial identity information, and an age classifier model as a discriminator to enforce the generator generated results to be more photo-realistic face images inside the objective age group. The objective function of IP-CGAN is a combination of three losses: an identity preserving loss that keeps the identity information of the generated image as close as possible to the original image, an age classification loss that enforces the generator to generate correct age group faces, and adversarial loss that enforces realistic image generation.

To find more natural-looking facial images, **Wavelet-domain Global and Local Consistent Age GAN (WaveletGLCA-GAN)** [52] model was proposed for age progression and regression of human faces. The architecture of the WaveletGLCA-GAN consists of a global network that catches the global structural information and three local networks that capture local texture descriptions of human faces. The objective function of the WaveletGLCA-GAN uses a combination of



Fig. 6. Samples generated with Stack-GAN. Image from Reference [55].

several losses such as an adversarial loss that make sure realistic wavelet generation, an identity-preserving loss that aims to preserve the face identity information, a pixelwise loss that reserve the input face image related to background information, age preserving loss that targets to improve the correctness of age synthesis, and the total variation loss that eliminates the ghosting effects.

Later, the WaveletGLCA-GAN model further extended to a Wavelet-GAN-based face aging method [53] that ensures the steadiness of features by inputting the facial feature vector and face image inputs to both the networks (G and D) and the output visually acceptable aged face image consequently. The face aging framework's architecture comprises a facial attribute embedded generator that learns the age mapping and the output face image with reliable characteristics and a wavelet-based discriminator that compares fake and real face images. This face aging framework uses three types of losses: an adversarial loss that advances the quality of generator generated images, a pixel loss that measures the image-level differences between the synthesized and the real images, and an identity loss that measures the feature-level differences between the synthesized image feature maps and the real image feature maps extracted by the VGG-Face descriptor.

4.1.6 Text-to-image Synthesis. Generating high-quality photo-realistic images from the text has tremendous computer vision applications, including photo-editing, computer-aided design, and image synthesis. **Attentional GAN (Attn-GAN)** [54] model was introduced for text-to-image translation. The Attn-GAN is a multi-stage generation model that progressively generates low-resolution to high-resolution images with m-number of generators. The Attn-GAN model consists of an attentional generative network that first draws sub-regions by applying conditions on given word descriptions that are mainly related to these regions and a multimodal similarity model that compute the resemblance between the synthesized image and the conditioned text. The Attn-GAN model's final objective function includes a deep attentional multimodal similarity model loss that computes the similarity between the synthesized image and the conditioned text. The GAN loss enforces the realistic image generation.

Stacked GANs (Stack-GANs) [55] proposed two-stage GANs for text-to-image translation by applying a condition on the given text descriptions via C-GAN [15]. The architecture of the Stack-GANs consists of two pairs of the GANs (GAN1 and GAN2), i.e., two GANs are set consecutively in the Stack-GANs for the generation of higher resolution image via text condition. The first pair of GAN at stage-1 produces a low-resolution image (64×64) by taking a noise vector conditioned to specified text information. The second pair of GAN at stage-II enhances the shortcomings of the first stage results and generates a high-resolution image (256×256). The objective function uses adversarial loss to generate a high-quality, diverse image. The Stack-GANs is the first work that can generate higher-resolution (256×256) diverse images, unlike the previous best GAN-based text-to-image models (see Figure 6).

Auxiliary Classifier GAN (AC-GAN) [56] is a semi-supervised text-to-image translation framework. The AC-GAN architecture contains a generator that uses a noise vector and class labels to generate a synthesized image and a discriminator that provides the likelihood that the input image belongs to the actual dataset and shows the class label's likelihood. The AC-GAN creates sharper, better quality, and high-resolution images and improves GAN training stability by preventing the mode collapse problem. Similarly, **Text Auxiliary Classifier GAN (TAC-GAN)** [57] is a semi-supervised text-to-image translation framework. The TAC-GAN concatenates the latent vector with the text embedding condition instead of a class label condition as AC-GAN does. The TAC GAN generator uses the textual description and noise-vector to generate synthesized data. The discriminator in TAC-GAN is similar to that in AC-GAN. The objective function of TAC-GAN uses an adversarial loss to successfully generate better quality images based on textual descriptions as compared to other advanced approaches.

Semantic Image Synthesis GAN (SIS-GAN) [58] model was proposed for text-to-image manipulation by employing text descriptions as conditions. The SIS-GAN generator follows the encoder-decoder structure where the encoder first encodes the input image into feature representation and then concatenates the feature representation with a text description. The decoder decodes the joint illustration into a synthesized image. The discriminator determines whether the input image is real and fits the definitions in the input text description. The SIS-GAN has quite specific and useful applications, like image manipulation systems.

4.1.7 Human Pose Synthesis. Pose transfer has many applications in the computer vision field; for instance, the movie-making character can be manipulated/estimated into the desired pose or the generation of the human pose's training data. **Pose Guided Person Generation Networks (PG²)** [59] framework used pose information explicitly synthesizing the person images in various poses. The PG² is a two-stage training process based on a coarse-to-fine fashion approach. Two GANs are set consecutively in the PG², where the stage-I generator uses the UNET architecture is also called Pose Integration Network, generates initial pose transferred results by capturing the global formation of a person's body conditioned on the target pose using masked L1 loss. The stage-II generator uses the UNET architecture called Image Refined Network, which generates sharper realistic images to bring stage-I results closer to the target image using adversarial loss. The discriminator at stage-II distinguishes synthesized and real pair of images.

Inspired by Ma et al. [59], a single-stage Deformable GAN [60] method was introduced for pose-guided human image generation. The Deformable GAN generates a person's image conditioned on appearance and poses information. To fix the pixel-to-pixel misalignments caused by the variations in poses between source pose (generated image) and target source (ground truth), the Deformable GAN used deformable skip connections to transfer correct details from a source pose to target pose. The nearest-neighbour loss is used as an alternative to the L1 and L2 losses to eliminate the different kinds of deformation between the fake and the real images and improve image generation superiority.

4.1.8 Stenographic Applications. Steganography is the collection of methods utilizes to hide secret information, e.g., a document, an image, or a video, within non-secret information. **Stenographic GAN (SGAN)** [61] and **Secure SGAN (SS-GAN)** [62] hide the secret information. The SGAN consists of three networks: (I) a generator network that generates realistic-looking synthetic images that could serve as secure containers for message embeddings, (II) a discriminator network that disintegrates between real and synthetic images; and (III) a Steganalysis classifier that determines whether or not realistic-looking images hide the secret details. The SS-GAN is similar to the SGAN model and uses the same technique to generate images with an adversarial learning scheme for steganography. The SS-GAN uses the same architecture as SGAN, but the only

difference between SGAN and SS-GAN is that SS-GAN is more reasonable for inserting messages with the arbitrary key than the SGAN, which is more feasible for adding messages with the same key, because SGAN becomes unreliable with the arbitrary key.

In recent years, Stegano-GAN [63] is the most popular approach used to hide arbitrary size binary data in images. The architecture of the Stegano-GAN consists of three networks: (I) an Encoder that takes an input image and secret message (binary message) of random size to generates the new steganographic image, (II) a Decoder that attempts to recover the embedded data (hidden message/binary message), and (III) a Critic that calculates the superiority of the original input image and the newly generated steganographic image. This critic network also provides feedback on encoder results, generates a more realistic image. Stegano-GAN objective function consists of the cross-entropy loss that measures decoding accuracy of the message, similarity loss that calculates similarity between steganographic and covers image, and realness loss of steganographic image using a critic network.

4.1.9 Image Manipulation Applications. In interactive manipulation, manipulates an image according to user goals. Tasks in applications of image editing contain to alter in color property or nature. **Interactive Generative Adversarial Network (IGAN)** [64] is a GAN-based image manipulation application where its interface allows the users to draw a rough sketch of the image. Then the rough outline of the image is used as input and produces the most similar realistic image. IGAN can automatically adjust the output while keeping users' edits as real as possible. IGAN changes the shape and color of the user sketch image by manipulating latent vectors that control the image's shape or color constraints. IGAN objective function contains a data loss that enforces user edits, a manifold smoothness loss that measure the distance between the latent representation vector and a new latent vector of user edits, and an adversarial loss that enforce the realistic image editing.

Text-Adaptive GAN (TA-GAN) [65] is another popular type of image manipulation application. The TA-GAN varies the specific visual attributes, like the input image's color or texture, according to the applied condition on given word-text descriptions. In TA-GAN, the discriminator creates word-level local discriminators according to applied word-text-description where each word-level local discriminator is attached to a specific kind of input image visual attributes, and the generator only modify the exact characteristics of the input image according to discriminator coarse training feedback while preserving the other text-irrelevant contents, like the layout and the pose of the image. The loss function of TA-GAN consists of a text-conditional loss, unconditional adversarial loss, and a reconstruction loss.

Similarly, users can make photo modifications as real as possible through interactive applications, so studying GAN creates different kinds of high-quality realistic images. **Introspective Adversarial Networks (IAN)** [66] is a GAN-based model introduced for image editing. They combine VAE [10] and GAN [1] frameworks. IAN is a custom-made user photo editing application called Neural Photo Editor. In Neural Photo Editor, a user can make any kind of adjustment on the given photo with the help of rough brush striking via the IAN interface, for example, painting in an area with green paint where the user wants to add green hair, and IAN rolls these rugged paint strokes into photo-realistic images that match the user's wishes. The objective function of IAN consists of a content loss that calculates the L1 loss between the reconstructed and real images, a feature loss that calculates the L2 loss between the reconstructed and real images, the ternary adversarial loss that enforces the generator to generate a more realistic image, and the KL divergence loss that impose high-quality image reconstruction.

Attribute GAN (Att-GAN) [67] is another facial characteristic editing framework that takes the facial feature information as a component of the latent vector demonstration that is a missing

gradient in previous facial editing applications. The Att-GAN consists of three essential components: (I) the classification component, which ensures proper manipulation of the attribute on the generated image, (II) the reconstruction component, which preserves the attribute-excluding information, and (III) the adversarial component, which is used for visually accurate generation. The final Att-GAN objective function uses an attribute classification loss, an adversarial loss, and a reconstruction loss to make the facial characteristic editing as real as possible. Dissection GAN (D-GAN) [68] framework introduced a new technique to visualize the inner working of a neural generator network and enable the users to not only edit the images, but a new image can also be added or removed from existing images according to user wish without any artistic skills. D-GAN comprises generators that represent all the necessary information to close the objective distribution, and the discriminator learns to catch the variance between real and fake images.

4.1.10 Visual Saliency Prediction. Visual saliency describes the region of interest in an image or video that attracts human attention. Extensive studies have been made on saliency detection to get accurate results. **Saliency Prediction GAN (Sal-GAN)** [69] was proposed for visual saliency prediction. The generator (Sal-GAN itself) predicts the saliency-maps from an input image's raw-pixels. The discriminator (refiner network) takes the output of the first to discriminate a saliency map into predicted or ground truth while updating the generator parameters. The objective function of Sal-GAN consists of a content loss that measure the distance between the projected saliency-map and the corresponding real saliency-map and an adversarial loss that enforces the predicted saliency-map toward the real saliency-map. The **Capsule-CGAN-based Saliency Detection (SalCapsule-CGAN)** [70] method integrates the popular capsule blocks [174] into both G and D instead of traditional UNET-like structural design. The generator predicts saliency-maps, which cannot be discriminated against the actual saliency maps, and the discriminator distinguishes the predictable output from the synthetic ones. The major drawback of the mentioned salient object detection method is that they cannot detect salient objects in a noisy scene.

Lately, **De-noising Saliency Prediction GAN (DSAL-GAN)** [71] detects the salient object in a noisy image. An end-to-end DSAL-GAN architecture comprised of two GAN: an image de-noising GAN and an object detection GAN. Image de-noising GAN cleans the noisy image. The generator of image de-noising GAN learns to generate a predicted image from the noisy input image. The discriminator of image de-noising GAN learns to categorize the predicted image from the actual one. Object detection GAN identifies the salient object in a de-noised image. The generator of object detection GAN predicts saliency maps by taking the output of image de-noising GAN (noise-free image) as input. The discriminator discriminates the predicted saliency map from the ground truth.

4.1.11 Object Detection. **Small object detection (SOD)** is a demanding task due to the small size and noisy version. The task of object detection becomes even more complicated when their appearance is invisible with other visible objects in the scene. **Segmentor GAN (Se-GAN)** [72] model detects the occluded objects in the same image. The Se-GAN segmentor network takes an image and visible area as its input and generates the mask of the whole object that has been occluded. The Se-GAN generator generates the appearance for the object painting's occluded area by painting the missing pixels. The discriminator of Se-GAN discriminates the generator generated and the actual image regions. Both the Se-GAN networks are trained in an adversarial way to generate an object image with invisible regions. **Perceptual-GAN (P-GAN)** [73] generates ultra-resolved descriptions of small objects for better detection by lessening the differences between small and large objects. The P-GAN includes a generator that transforms the small objects' sparse representations to highly super-resolved images that are sufficiently like actual large objects and a perceptual discriminator that differentiates the generator-generated super-resolved

representations of small objects from the real through an adversarial loss. In addition to that, the discriminator network boosts the detection performance through an additional perceptual loss.

Similarly, **Multi-Task GAN (MT-GAN)** [74] used a super-resolution network to up-scale the small-scale distorted image into the large-scale clear image for better detection. It consists of a super-resolution network and a multitask network. MT-GAN's super-resolution network works as a generator, which up-scales the small-scale distorted image into a large-scale clear image. MT-GAN's multitask network works as a discriminator to discriminate the real higher resolution images from those generated, predict object categories scores, and further improve the bounding boxes at once. Recently, the **GAN-based Detection of Objects (GAN-DO)** [75] method learns an adversarial objective for object detection through training. GAN-DO takes a low-quality image as input for accurate object detection compared to previous object detection methods that take a high-quality image as input. The discriminator learns to differentiate between the output of higher quality original data from the pre-trained baseline model and the generator's different quality output. The generator learns to outsmart the discriminator. The discriminator classifies the generator output of the augmented data as the output of the original data by the baseline model.

4.1.12 3D Image Synthesis. Advances in 3D volumetric convolution networks have led to GAN application to generate 3D objects. The **3D GAN (3D-GAN)** [76] expands the system of 2D-GAN by one more dimension to address three-dimensional data, i.e., 3D-GAN. The 3D-GAN generator has five volumetric FC layers (kernel sizes of $4 \times 4 \times 4$) with strides of 2. The 3D-GAN discriminator uses Leaky ReLU, while it is basically like the 3D-GAN generator. The 3D-GAN has achieved some actual incredible results as compared to previous advanced 3D models. However, there are significant limitations in using 3D-GAN in high memory, high computational costs, and limited to only single view rendered image reconstruction. A noticeable fall in performance occurs when used for non-rendered images. Similarly, **Projective GAN (Pr-GAN)** [77] inferred 3D objects from multiple 2D views of an object. They used a 3D volumetric convolution network to generate 3D shapes in the generator. They used a 2D projected image as an input of discriminator to match synthesized 3D objects (fake objects) with the 2D view. They also introduced a 3D-VAE-GAN expansion to 3D-GAN by including an image encoder network, which outputs the latent vector representation by taking a 2D image as input.

Furthermore, the **3D-CGAN** [78] used paired training data to generate 3D-samples in various and customizable rotation angles. The generator of 3D-CGAN takes a noise vector and the condition as a combined input by keeping the input vector similar and varying the condition for the generation of related and reasonable examples in diverse conditions for the identical input vector. The generator output is a (32^3) 3D matrix. The 3D-CGAN discriminator discriminates the real and fake samples. **Tree-GAN** [79] proposed a novel computationally efficient tree-structured graph convolution network for 3D point cloud generation of multi-class applications in an unsupervised manner. The Tree-GAN generator uses a single point as an input from a Gaussian distribution and produces 3D point clouds as an output. The Tree-GAN discriminator forces the generator to produce more real points by differentiating the real and generated point clouds. The **3D GAN (3D-GAN)** [80] provides 3D object generation and 3D object recognition models from a latent vector using the advances in new convolution networks. The generator maps 200 shaped noise sampled at random to a cube of $64 \times 64 \times 64$, representing a subject in the voxel, volumetric pixel space. The discriminator will give the likelihood of the object input being real or fake.

4.1.13 Medical Applications. GAN-based medical image applications are a well-accepted active area of research. **Segmentation GAN (SegAN)** [81] proposed a segmentor-critic arrangement for medical image segmentation where a critic network is used instead of the discriminator and the segmentor instead of the generator. They used the critic network, because a regular discriminator's

single value result may not be sufficient for dense, pixel-level segmentation of medical images. A fully CNN-based segmentator as a generator generates segmentation label maps. The SegAN segmentor generates a projected segmented image, and the SegAN critic optimizes the differences in the hierarchical structure between the segmentation generated image and the actual image. **Medical GAN (Med-GAN)** [82] generates realistic **electronic health records (EHRs)** by using Med-GAN based on the PAMF and the MIMIC-III datasets. For EHRs, the author uses a hybrid technique by combining an AE [9] with GAN [1], because GAN cannot handle high-dimensional multi-label discrete variables well. Med-GAN takes EHRs dataset as input and generates high-dimensional discrete variables. Med-GAN-created EHRs play a vital role in expert medical opinion.

Additionally, GANs have also been used in many other medical applications such as **drug discovery GAN (Chem-GAN)** [83] to determine drugs' inner element diversity. In this article, the author introduces an evaluation metric that computes the model's inner element diversity generated molecules and natural molecules, respectively. **Feedback GAN (FB-GAN)** [84] produces and design synthetic deoxyribonucleic acid (DNA) sequences coding for **antimicrobial peptides (AMP)**. The FB-GAN consists of a GAN that generates novel raw gene sequences and an analyzer, which converts a gene sequence into a probability that it encodes AMP. GANs can generate new DNA components, genes, or even regulatory circuits with attractive functions. GANs can also be trained to recreate images or brain activity measurements directly from fMRI responses on real data with data augmentation from a simulated encoding model [85]. Here, GANs are trained to learn complex image reconstruction from noisy fMRI images, representing brain activity. GAN-based teeth restoration system [86] proposed to restore the missing tooth crown. The generator of the teeth restoration system uses the UNET architecture with symmetric skip connections. The UNET contains an encoder-decoder structural design. The generator prepared the missing tooth crown depth image. The teeth restoration system's discriminator classifies the crown missing depth image (predicted image) from the crown-filled depth image (the ground-truth).

GAN-based online doctor recommendation framework (Dr-GAN) [87] introduces to find a suitable doctor for the treatment of their health based on on-line questions and answers. The Dr-GAN is a doctor-patient interaction system in health-care service. The Dr-GAN system helps select the best doctor to handle the disease or diagnose it without any personnel assistance. A novel framework based on GANs, named Med-GAN [88], was proposed for medical I2I translation tasks. The Med-GAN-based approach's fundamental objective is to perform image translation tasks between motion-corrupted images to their corresponding motion-free images but limits their work to 2D slices. MedGAN consists of a CasNet generator that follows the structural design of the encoder-decoder, which translates source domain input images into the target domain, a discriminator that distinguishes the actual and transformed images, and a pre-trained characteristics extractor used to extract vibrant characteristic to measure style transfer losses.

4.1.14 Facial Makeup Transfer. In facial makeup transfer, one face's makeup stylishness is transferred to another without losing the non-makeup face's identity, and the makeup removal system does the reverse. A novel dual input/output GAN called **Beauty GAN (B-GAN)** [89] transfers the makeup style from a makeup face to another non-makeup face on the instance-level. B-GAN architecture consists of one generator network (G_1) and two discriminator networks (D_1 and D_2). The generator network takes the makeup and anti-makeup face images as inputs and produces a new after-makeup face and an anti-makeup face. The discriminator network (D_1) discriminates the generator-generated makeup face image from the real makeup samples dataset. The discriminator network (D_2) discriminates the generator-generated anti-make-up face image from the real non-makeup samples dataset. B-GAN's objective function contains four types of losses: makeup constrains loss, adversarial loss, cycle consistency loss, and perceptual loss.

Paired CycleGAN [90] is an unsupervised learning approach, introduced a makeup transfer and makeup removal system by coupling two separate networks (GANs). The first network transfers a variety of arbitrary makeup styles. The second network does the inverse with the end goal: the output of their successive applications matches the input. Both the networks (both the GANs) used an adversarial way for joint training. The final loss of Paired-CycleGAN includes a style loss, an identity loss, and an adversarial loss. However, this approach does not work as well on severe makeup-styles unseen during training. Inspired by Reference [89], the **Disentangled Makeup Transfer GAN (DMT-GAN)** [91] model was proposed for facial makeup transformation where the input image first decomposed into two independent parts, the makeup-style and personal identity parts, and then the encoders-decoder mechanism is applied (one encoder for personal identity, second encoder for makeup style, and the single decoder for the reconstruction of the original image). The DMT-GAN discriminator makes a distinction between generated and real images. The objective function of DMT-GAN uses a mixture of several losses, such as the makeup loss, perceptual loss, identity makeup reconstruction loss, reconstruction loss, attention loss, adversarial loss, KL loss, and total variation loss, to succeed in diverse scenarios of makeup transfer.

4.1.15 Facial Landmark Detection. The objective of facial landmark detection is to identify human faces with different vital points. Landmark detection has various facial re-enactment applications, head pose estimation, face recognition, and 3D face reconstruction. **Style-Aggregated Networks (SAN)** [92] proposed an idea for facial landmark detection. The SAN is a two-stage training process: a style aggregated face generation model and a facial landmark prediction model. The style aggregated face generation model of SAN transforms the input image into various styles and then fuses them into a style aggregated face. The generated style-aggregated face image (mean face) covers the significant variance in image styles. The facial landmark prediction model of SAN takes the style aggregated face and original face as input to obtain two complementary features representing a different kind of useful information. Then a cascade technique was used to produce predictions of the heat map. The SAN model can successfully handle the significant intrinsic discrepancy of image styles.

Another GAN-based face landmark locations framework [93] was used to retrieve uniform face landmarks as an attribute for a simple **Support Vector Machine (SVM)** classifier for correct classification between real and synthesized fake human face images. The extracted normalized facial landmark points used as an attribute vector for an SVM classifier get good competitive results compared to the best methods for the GAN-based fake images of the human faces. This SVM classifier-based method tries to determine that facial landmarks of the generated human face image by the GANs have different configurations from the real human face image due to the lack of global constraints.

4.1.16 Image Super-resolution. Image **super-resolution (SR)** aims to up-scale the images of low-resolution, because images with higher resolution have better visual qualities. Super-resolution image has been commonly used in satellite, medical, military, and many more tasks. **Super-Resolution GAN (SR-GAN)** [94] was introduced to improve image resolution. The SR-GAN takes an image of low-resolution and induces an image of higher resolution through $4 \times$ up-scaling factors. The SR-GAN model uses the deep residual network as a generator and pre-trained VGG-19 [205] as a discriminator. The SR-GAN generator takes the low-resolution input images and derives the photo-realistic higher resolution images. The discriminator network of SR-GAN differentiates the generator-generated super-resolved images from original photo-realistic images. The SR-GAN model's cost function uses a perceptual loss that measures the feature-level differences between the synthesized and the real images feature maps extracted by the VGG-Network and the adversarial loss that pushes the generator-generated images toward realistic images.

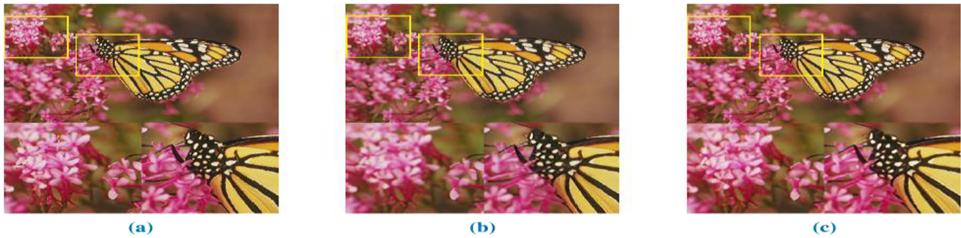


Fig. 7. (a) Real image, (b) SR-GAN result, and (c) ESR-GAN result are shown in the figure, which shows that ESRGAN delivers better visual performance than SRGAN. Images from Reference [95].

The SR-GAN-generated texture information is not sufficiently real due to noise. **Deep Enhanced Super Resolution GAN (ESR-GAN)** [95] further improves the visual quality results of the SR-GAN model by making three significant changes to the overall structure of the SR-GAN model, such as the use of Residual-in-Residual Dense Block that has a superior learning power and more comfortable to train, Relativistic GAN [220] improves the classification capacity of the discriminator and enhances the perceptual loss through VGG-19. Taking advantage of these changes, the ESR-GAN model achieves more significant visual quality results with more photo-realistic looking textures than the SR-GAN and received the first prize with the highest perceptual directory in the challenge of PIRM 2018-SR (see Figure 7).

Furthermore, **Super Resolution Dual GAN (SR-DGAN)** [96] proposed that learns to solve the noise before the super-resolution problem for the generation of the noise-free high-resolution images. The architecture of SR-DGAN consists of two sub-networks: (1) a high-to-low-resolution image network and (2) a super-resolution network. A high-to-low-resolution network of SR-DGAN generates a pair of natural-looking images, trained with mobile captured real-world images. A super-resolution network (low-to-high resolution network) of SR-DGAN generates the super-resolution image. The SR-DGAN shows better performance than SR-GAN and ESR-GAN models. **Deep Tensor GAN (DT-GAN)** [97] model cascades tensor structures, i.e., a brief illustration of the input image with DC-GAN [16] architecture for the generation of a high-quality large-size realistic image. The working mechanism of DT-GAN consists of a training phase and a generation phase. The DT-GAN training phase trained the DC-GAN model with low-resolution images as input to generate image tensors of low-resolution through latent vectors. The generation phase of DT-GAN generates LR tensor images from the latent vectors via trained DC-GAN. The DT-GAN can generate better quality images, particularly big-size images.

4.1.17 Texture Synthesis. Texture synthesis is a traditional and exemplary issue in the image domain to generate a new texture indistinguishable from the real texture. **Markovian GAN (M-GAN)** [98] is a texture synthesis technique based on GAN [1] with the vital dissimilarity of just working on neural patches despite full image. The M-GAN generator takes the content image feature-maps as its input and generates a new texture image. The M-GAN discriminator learns to discriminate the real feature patches from inappropriately synthesized ones. The objective function of M-GAN includes a feature loss and a texture loss. The M-GAN has a low computational cost with real-time achievement to synthesize neural texture and stylized videos compared to the primary texture synthesis method [99].

Spatial GAN (S-GAN) [100] maps a spatial tensor of an image is the first edition of GAN in texture synthesis with completely unsupervised wisdom and well appropriate for arbitrary large size texture synthesis. The S-GAN uses the same network architecture as DC-GAN [16]. The S-GAN generator transforms a spatial noise-array from a prior distribution into an image data-space via a



Fig. 8. T-GAN generated samples in the second row from first-row input sketches. T-GAN can produce new examples of usual things from hand-drawn sketches and texture patches [102].

fractionally stride-CNN stack. The S-GAN discriminator uses a stack of CNN to categorize whether an image is a generator-generated image or real. The S-GAN model has significant achievements, like generating high-quality textures of any desired size with a single forward pass. However, the S-GAN cannot handle all types of textures, and there is also no good morphing texture.

Similarly, Periodic Spatial GAN (PS-GAN) [101] further extended the S-GAN model to synthesize different, cyclic, diverse, high-resolution, and complex datasets texture. The PS-GAN model has some significant achievements like it can handle textures of vast vulnerability from big-size images; it can generate high-quality textures of any wanted size with a single forward pass of a CNN. The objective function of PS-GAN uses an adversarial loss to generate realistic-looking natural texture. However, the PS-GAN can suffer from a “mode dropping” problem, and it has an unstable convergence mechanism.

4.1.18 Sketch Synthesis. Sketch representations are a constructive way to draw what users need. **Texture GAN (T-GAN)** [102] method was proposed for sketch synthesis that directly derived and extended from the popular Scribbler method [103] with the new authorized-control over the textures of the object. T-GAN generator takes the texture, sketch, and color images as inputs and generates an image as output (see Figure 8). The discriminator of T-GAN discriminates the generator-generated results from real ones. The objective function consists of four types of losses: feature, content, adversarial, and texture losses. The texture loss forces the texture look of the generated image close to the real image, while the remaining three losses are defined similarly to the popular Scribbler method. The concept of a binary mask is also used in T-GAN to impose the computing content loss, and texture loss only works in the foreground area.

Like T-GAN, [104] introduced a CGAN-based sketch-to-image translation auto-painter method that automatically generates compatible colors for cartoon images. Their framework converts the sketches to cartoon images by adopting the network architecture of pix2pix [27]. The generator considers the sketch as input and generates the same resolution colorful cartoon image as output. The discriminator classifies the input sketch-image pair is either fake or real. The cost function uses four losses, such as a L1 distance loss that calculates the difference at pixel level between the generated and the real images, an adversarial loss that encourages more deviations and colorfulness in the generated images, a feature loss (L2 distance loss) that calculates the difference at feature level between the generated and the real images, and a total variance loss that encourages smoothness in the generated images.

Sketchy-GAN [105] is a simplified sketch-to-image translation process that takes the input sketch. The output is a photo-realistic image with the same pose from 50 diverse classes, including motorcycles, horses, and couches. The Sketchy-GAN generator uses an encoder-decoder architecture that takes the input sketch and generates a photo-realistic image of a similar pose. The discriminator of the Sketchy-GAN categorizes whether the given image is fake or real.



Fig. 9. Some experimental results of Cycle-GAN [108].

The classification results of the discriminator enforce the generator to produce more realistic images. The Sketchy-GAN uses the same loss functions as pix2pix [27]. Most recently, the **Composition-Aided GAN (CA-GAN)** [106] framework was proposed for face photo-sketch synthesis for paired training data of photo/sketch and the related face labeling masks. The CA-GAN generator takes a face mage (photo/sketch) and the related face mask as inputs and generates a new face image (photo/sketch). The CA-GAN discriminator discriminates whether the given photo/sketch image is a fake or real image. The CA-GAN's final objective function uses a mixture of the composition loss function that prevents the huge components from overpowering the generator during training, the perceptual loss function that encourages the generated and real images to be perceptually alike, and the adversarial loss function that encourages realistic images generation. The CA-GAN output is further refined to get sharp results through Stacked Composition-Aided GAN, which follows the two stages training process, is trained together.

4.1.19 Image-to-image Translation. The purpose of the **image-to-image translation (I2I)** methods is to renovate an image's visual illustration with the new visual representation. The GAN-based pix2pix [27] method adopts a supervised learning technique for I2I translation. The generator of pix2pix translates the source image into the target image based on the condition applied, and the discriminator makes it confirm whether the used condition is meet-up by considering the pixelwise loss. The architecture of pix2pix-GAN uses adversarial loss that enforces the generator to produce a more realistic image and L1 distance loss that calculates the distance between the fake and the real images. Similarly, **Perceptual Adversarial Networks (PAN)** [107] introduced another method for I2I where the perceptual loss is minimized instead of minimizing the pixel loss between paired images. The generator (transformation network) of PAN reduces the discrepancy that the discriminative explored. PAN's discriminator enforces the predicted image is perceptually similar to a real one by minimizing the perceptual information dissimilarity. The objective function of PAN is composed of the adversarial and the perceptual losses. Cycle-GAN [108], **Discover cross-domain relations with GAN (Disco-GAN)** [109], and **Dual GAN (Dual-GAN)** [110] introduced I2I methods for unpaired training data. The Cycle-GAN presents a cycle-consistency loss by coupling two different pairs of GANs together. The first pair of GANs translates the image into another domain, and the second pair of GANs performs the inverse without paired training data. It is proved that cycle-consistency loss with adversarial loss can protect the actual image after a cycle of forwarding and backward translations. The Cycle-GAN architecture prevents generators from extreme hallucinations and collapsing mode by enforcing cycle-consistency mechanisms. So, the cycle consistency loss forces an image to be rendered back to itself after the translation cycle, and adversarial loss takes a new cost function where sigmoid cross-entropy loss is replaced with a least-square loss for stable training plays a significant role in escaping the mode collapse. Cycle-GAN produce more realistic images as shown in Figure 9.

The Disco-GAN discovers the cross-domain relations between different domains given unpaired training data. The Disco-GAN has nearly the same basic idea as Cycle-GAN. It is proved that discovered relations can successfully transfer the image style from source to destination while preserving the original image's crucial attributes. The Disco-GAN can generate high-quality images with transferred style from one to another, but it may hurt artifacts at the boundaries. Like previous unpaired image translation methods, Dual-GAN has almost the same architecture as cycle-consistent adversarial networks (Cycle-GAN). The Dual-GAN method uses Wasserstein distance instead of a sigmoid cross-entropy loss and cyclic reconstruction loss. Dual-GAN generates good results than Cycle-GAN and Disco-GAN. StarGAN [111] is another unsupervised I2I method that uses a single model for multi-domain I2I translation. StarGAN architecture consists of a generator and a discriminator. The StarGAN generator takes the image and the domain information (labels indicate the domain information) as input and translates the image into the appropriate domain. The StarGAN discriminator discriminates against the real image from the generated image and categorizes it to its related domain. The architecture of StarGAN uses three types of losses, such as adversarial loss, reconstruction loss, and classification loss. The StarGAN has promising results on performing expression, gender, age, and facial attribute transfer tasks. **Unsupervised I2I Translation (UNIT)** [112] model was proposed for I2I, which combines VAE and GAN architectures. UNIT fuses the GAN [1] for generating the corresponding image in two domains through shared-latent space constraints and the VAE [10] for an edited image with the input image in the relevant fields. The UNIT performs the I2I from one domain to another without any consequent images in the training dataset.

To deal with this curb, **Multi-modal UNIT (MUNIT)** [113] generates different results from a given basis domain image. The MUNIT first decomposed the image's latent space into style, capturing the domain-specific properties and content code, domain-invariant. For I2I translation from one domain to another, they re-combined the content code with a random style code in the target style space. Better results in quality and diversity against existing advanced methods for translation of images show its superiority. Diverse I2I Translation via **Disentangled Representations I2I (DRIT)** [114] model is a multi-modal generation framework that can generate different results without paired training data. The DRIT purpose-specific disentangled representation, which enables the efficient generation of multi-modal outputs. The DRIT uses adversarial content loss and the cross-cycle consistency loss for multi-modal mapping between the different domains with non-paired data. The limit of the DRIT is that they fail when the source domain characteristics and target domain characteristics vary significantly. Likewise, DRIT++ [115] expands the DRIT model by (I) integrating sample diversity mode-seeking maximum likelihood and (II) generalizing the two-domain structure to tackle multi-domain I2I issues.

4.1.20 Face Frontal View Generation. The enormous pose disparity among face images is a significant obstacle in the computer-based face recognition system. For an automatic face recognition system, frontal face activity is performed to change a face image from various perspectives to frontal perspectives for better outcomes. Disentangled Representation Learning GAN (DR-GAN) [116] introduced a face frontal view generation model for the pose-invariant face recognition system by modifying the traditional GAN discriminator with a multi-task discriminator. The generator of DR-GAN uses the encoder-decoder structural design. The generator encoder represents the identity attribute, and the decoder generates an image by following the encoded identity and the pose information. The multi-task discriminator of DR-GAN classifies the fake and the real images but performs identity and poses classifications for real images and fake images. DR-GAN objective function includes an adversarial loss that generates a pose-invariant face recognition system. DR-GAN demonstrates better results than other pose-invariant face recognition methods.

Two Pathway GAN (TP-GAN) [117] proposed a natural-looking frontal view generation model from a single face view image. The TP-GAN generates an identity preserving image after analysis of the facial-appearance in detail. The TP-GAN architecture comprises a global generator that generates the face's global structures, a local generator that generates the appearance details around facial landmarks, and a discriminator that discriminates against the generator generated frontal views from real frontal views. TP-GAN objective function includes a symmetry loss that enforces the generated face image to be horizontally symmetric, a content loss that calculates the dissimilarity between the fake and the real face images, an identity-preserving loss that guarantees the identity of the generated image, an adversarial loss that enforces realistic image, and a total-variation loss that removes the ghosting effects.

Face Frontalization GAN (FF-GAN) [118] is another face frontal view generation model. The FF-GAN blends 3D Morphable Model [119] into GAN formation to give appearance and shape priors for face images. The FF-GAN architecture contains a generator that takes an image of a non-frontal face as input and generates a frontal face image and a discriminator that discriminates the real frontal face image from a generator-generated one. The objective function comprises a symmetry loss, a content loss, an identity-preserving loss, an adversarial loss, and a total-variation loss. Additionally, to protect the new frontal face image's face identity, a regularizing term is added. The FF-GAN takes the benefit of quick convergence with a small dataset and generates high-class frontal face images.

To deal with complicated poses and part occlusion, **GAN-based Face-Transformation with Key Points Alignment (FT-GAN)** [120] proposed the front face alteration method. The FT-GAN is a learning process involving two tasks where the first task uses per-pixel transformation phenomena through Cycle-GAN architecture [108] for frontal face synthesis, and the second task uses crucial point alignment for refining the transformed face. The FT-GAN architecture contains a generator composed of two stride-2 convolutional layers, nine ResBlocks, two stride-12 fractionally strided convolutional layers, and a discriminator classifying a patch (70×70) in an image as an actual patch or generator. The final loss of FT-GAN consists of an adversarial loss that enforces the realistic image generation, a cycle consistency loss that enforces an image to be translated back to itself for the generation of the frontal face with the promising rate of success., a pixel-to-pixel transformation loss that calculates the difference between synthesized frontal face image and initial non-frontal face image, and a key point alignment loss that calculates the difference between synthesized frontal face image and initial non-frontal face image.

4.1.21 Others Image Domain Applications. Moreover, GANs have been broadly used in many new image processing applications other than the applications discussed in the article. The list of these GANs-based applications is given in Table 5.

4.2 Audio Domain

4.2.1 Language and Speech Synthesis. GANs have excellent success in synthesizing data in music generation, dialogue systems, and machine translation. **Ranker GAN (Rank-GAN)** [144] proposed a novel high-quality language (sentence) generation method. They employed a ranker network instead of the traditional discriminator network. The applied ranker network achieved exceptional performances, because the traditional discriminator only does best in binary classification. Through adversarial training, the Rank-GAN generator tries that its generated sentence/language sample (machine written sentences) is so realistic that it ranks higher than a real sentence (human-written sentence). Simultaneously, the Rank-GAN ranker network calculates the actual sentences' ranking score superior to the generated sentences. A voice conversion system converts the source voice to a target voice without changing its linguistic contents. **Variational**

Table 5. List of other GAN Exciting Applications

#	Model Name	Goal
1.	Cycle-Dehaze [121]	Image de-hazing.
2.	Disentangled-Dehazing [122]	
3.	Cycle-Defog2Refog [123]	
4.	Alpha-GAN [124]	Image matting.
5.	Pan-Sharpening GAN [125]	Images fusions.
6.	Task-Oriented GAN (TO-GAN) [126]	Image classification.
7.	Pixel-DTGAN [127]	Cloth image.
8.	Gene-GAN [128]	Object transfiguration.
9.	Semi-Supervised GAN (Ss-GAN) [129]	Semantic segmentation.
10.	Face-Mask-CGAN [130]	Portrait editing.
11.	Vital [131]	Object tracking.
12.	Sample-Level GAN (SL-GAN) [132]	
13.	Semi-Latent GAN (SL-GAN) [133]	Face image generation.
14.	Semantically Decomposing GAN (SD-GAN) [134]	
15.	Wav2Pix [135]	
16.	GAN-Fit [136]	
17.	Deforming Autoencoder [137]	
18.	Dual Variational Generator [138]	
19.	3D Aided Duet GAN [139]	
20.	Pose-Normalized-GAN (PN-GAN) [140]	Person re-identification.
21.	Identity-Preserving GAN (IP-GAN) [141]	
22.	Deep Convolutional GAN [142]	De-occlusion the image.
23.	UNET & GAN [143]	

Auto-encoding WGAN (VAW-GAN) [145] proposed a voice conversion system by combining the VAE [10] and W-GAN [20] frameworks together. The VAW-GAN framework improves the target results with a realistic spectral shape. In VAW-GAN, the encoder provides the source voice with a phonetic substance, and the decoder integrates the transformed target voice with the information provided by a target speaker.

4.2.2 Music Generation. GANs have also been utilized for music generation, like **Continuous Recurrent Neural Network GAN (C-RNN-GAN)** [146] generates continuous sequential data. The C-RNN-GAN model both the networks (generator and discriminator) as a RNN (recurrent neural network) with LSTM (long short-term memory) [147], precisely obtaining the full sequences of music. They have bad results, because the distinct value of the music elements has no consideration, i.e., evaluating the entire sequences is possible, but evaluating a partially generated sequence is not possible. To have better outcomes, **Sequence GAN (Seq-GAN)** [148] and **Object Reinforced GAN (OR-GAN)** [149] consider the sequence generation method as a chronological decision-making procedure by considering the distinct property of the generated music. They performed a policy gradient algorithm by not producing the entire sequence at once, because a traditional GAN cannot produce discrete output due to the gradient flow of the discriminator to the generator, resulting in small changes in the result of the generator instead of the output proper to the discrete data space. They treat a generator's output as an agent policy and take the discriminator's output as a reward. Choosing a discriminator's reward is a normal choice, as the generator works to achieve the discriminator's reward, similarly to an agent seeking to get a significant

Table 6. List of other GAN Exciting Applications

#	Model Name	Goal
1.	Utility-GAN (U-GAN) [150]	Question-answer selection.
2.	Multi-Scale Matching GAN (MSM-GAN) [151]	
3.	Resume-GAN [152]	Automatic matching CV for talent-job fit.
4.	Natural Language Processing GAN (NLP-GAN) [153]	Review detection and generation.
5.	Fake-GAN (F-GAN) [154]	
6.	Information Retrieval GAN (IR-GAN) [155]	Information retrieval.
7.	Personalized Search GAN [156]	
8.	Adversarial Reward Learning GAN (ARL-GAN) [157]	Visual storytelling.
9.	Speech Enhancement GAN (SE-GAN) [158]	Speech enhancement.
10.	Frequency Domain SE-GAN (FDSE-GAN) [159]	Speech recognition.
11.	Caption-GAN (C-GAN) [160]	Image caption.
12.	Image-to-Poetry GAN (I2P-GAN) [161]	Poetry generation.

reward in reinforcement learning. OR-GAN differs significantly from Seq-GAN, incorporating a hardcoded cost-function to the reward operation to achieve a particular objective.

4.2.3 Others Audio Domain Applications. Furthermore, GANs have been broadly utilized in many other music generations, dialogue systems, and machine translation applications other than the applications discussed in the article as shown in Table 6.

4.3 Video Domain

4.3.1 Video Applications. Here, we discuss video generation and prediction GAN. The video is mostly a combination of stationary background and moving objects, where predicting the object's motions is a core issue in computer vision. The **GAN-based video generation (V-GAN)** [162] technique decomposes the video frame into the content and motion parts. The V-GAN contains two generators (G_1 and G_2), one for moving foreground and the second for the static background. A moving foreground generator (G_1) generates the 3D video and the related three-dimensional (3D) masks for the foreground, with spatial-temporal 3D CNN, and a static background generator (G_2) generate the background image with 2D CNN. The discriminator takes the generator's complete video as input and tries to discriminate against actual videos. The objective function of V-GAN uses adversarial loss to generate successfully future prediction videos at a favourable rate. V-GAN can generate small-duration videos, but it requires a large size memory.

Later, **Motion and Content GAN (MoCo-GAN)** [163] introduced a video generation framework. The proposed framework generates the video in an unsupervised way by decomposing the video into motion and content part of the latent space. To generate a video frame, the decomposed content and movement parts are fed into the generator. An image discriminator (D_1) discriminates the real frames from generator-generated frames. A video discriminator (D_2) discriminates the real videos from generator-generated videos. The MoCo-GAN model uses adversarial loss for video generation at an encouraging rate of success. Recently, the **Disentangled Representation Net (DRNET)** [164] approach learns disentangled image representations from GAN-based video. The DRNET architecture consists of two encoder networks that produce distinct attribute representations of content and pose. One decoder network predicts the future frames after receiving the concatenated results from encoders. The architecture of DRNET uses three types of losses such as reconstruction loss that measures the dissimilarity between the generated and

Table 7. List of other GAN Exciting Applications

#	Model Name	Goal
1.	Pose-GAN (P-GAN) [166]	Future frames/events prediction.
2.	Dual Motion GAN (DM-GAN) [167]	
3.	Recycle-GAN [168]	Video retargeting.
4.	Video Q & A via Multi-Modal CGAN [169]	Video question answering the given question.

the real images, similarity loss that measure the structural difference, and adversarial loss that enforces the generator-generated results toward ground truth.

Dual Video Discriminator GAN (DVD-GAN) [165] framework generates high-resolution videos of comparatively high reliability. The DVD-GAN built upon the advanced BigGAN [24] is composed of one generator (G) and two discriminators ($D1$ and $D2$). The DVD-GAN generator learns to generate higher resolution photo-realistic scenes and temporal dynamics. However, video prediction requires a generator to explore the conditioning frames and determine the scene elements, which advances over time. The DVD-GAN discriminator ($D1$) reviews only a frame content and organization, while the discriminator ($D2$) should give a generator with the learning-sign to generate movement. The DVD-GAN can capture the density of a large video dataset such as UCF-101 and KINETICS-600 datasets.

4.3.2 Others Video Applications. Moreover, GANs have been broadly used in many other new video applications other than the applications discussed in the article. The list of these GAN-based applications is given in Table 7.

4.4 Climate and Earth Science Domain

GAN's acceptance and significance have led to several practical applications, which have some significant effects on human society, such as earth & climate changes. GAN's recent development further expands its utilization areas directly related to human life, such as speedy and accurate future typhoon predictions, which can reduce the world's damage by taking the earlier precautionary measures.

A novel **GAN-based State-Parameter Identification model (SPID-GAN)** [170] was proposed to concurrently learn the bi-directional mappings between model parameters and model states of a geophysical model at a single timestep. The architecture of the SPID-GAN follows the Cycle-GAN [171] framework, consists of the two generators and two discriminators trained together. The first pair of GAN used to identify the forward parameter-to-state mapping (i.e., forward mapping of model state space from parameter space). In contrast, the second pair of GAN used to identify the reverse state-to-parameter mapping (i.e., a reverse mapping from the model state space to the parameter space). The cycle-consistency between two mappings alleviates the mode collapse problem by reducing the reconstruction loss. The SPID-GAN framework has vast applications in the geophysical research community, like measuring the groundwater flow.

Recently, a novel GAN-based 4D-Seismic-Inversion model [172] was proposed to reverse impedance change images for reservoir property changes. The architecture of the 4D-Seismic-Inversion model follows the Cycle-GAN [108] framework to simultaneously learn the bi-directional mappings between the reservoir dynamic property changes and seismic feature changes, consists of two GAN pairs trained jointly, where each pair of GAN share the similar network architecture, is responsible for learning the mapping from one domain to another. The first pair of GAN used to identify the forward reservoir dynamic property changes-to-seismic attribute changes mapping (i.e., forward mapping from the reservoir dynamic property changes

and seismic attribute changes). The second GAN pair used to identify the reverse seismic attribute changes-to-reservoir dynamic property changes mapping (i.e., a reverse mapping from the seismic attribute changes to reservoir dynamic property changes). Although the 4D-Seismic-Inversion approach gives the best results in image analysis, a massive amount of previous research is required to prepare the synthetic training datasets. The 4D Seismic Inversion has vast applications, like water breakthrough risk calculation before drilling.

Furthermore, Rüttgers et al. 2019 [173] proposed a typhoon prediction method with time-series of past typhoons satellite images as input, and the output images show the typhoon hours ahead. The engaged GAN-based model's output images (predicted typhoon images) can help identify the typhoon center location and the future's distorted cloud structures. For this, the author first trained the GAN-based model on typhoon satellite image datasets, and then the trained GAN-based model is used to generate a 6-hour earlier typhoon track for which the model is not trained earlier. The difference between the generated typhoon images and the actual typhoon is calculated in kilometers. These speedy and accurate future predictions will help reduce the world's damage by taking the earlier precautionary measures. Like the typhoon track prediction, the GAN-based model can detect flood or river stream flows. Qian et al. 2019 [174] proposed a data-driven model for flood monitoring, prediction. The CGAN-based flood prediction model take-out the dynamics of flood from simulated-data. The author employs the UNET with skip connection as a generator, which generates the fake flood, and the patch discriminator discriminates against the fake-flood from the real-flood. Performance evaluated in terms of MSE and PSNR results shows that the flood prediction model can offer instantaneous flood exploitation predictions.

4.5 Miscellaneous GAN Applications

GANs have also been used in variety of other domains such as computer security [175], chess game playing [176], network pruning [177, 178], spatial representation learning [179], mobile user profiling [180], data augmentation [181, 182], heterogeneous information networks [183], privacy-preserving [184–187], social robot [188], cipher cracking [189], autonomous driving [190, 191], continual learning [192], particle progress in oncology [193], GANs for economic and financial management [194–199], GANs for textile goods [200, 201], GANs for fluid-flow [202, 203], GANs for fires & earthquake prediction [204, 205] and many others in dominant areas of research.

5 GAN TRAINING

This section will survey several training obstacles associated with GAN training and several training techniques to improve GAN training to generate more realistic data.

5.1 Problems with GAN Training

GANs are influential generative models but deeply hurt from unstable training due to several GAN training challenges. In this section, we review some of the training obstacles for detailed discussion. Supplementary Material provides an analysis to understand better the training issues' main reasoning with the appropriate remedies discussed in this survey.

5.1.1 Nash Equilibrium. GAN training may be considered two deep neural networks, competing for one against the other in an adversarial way to search for **Nash equilibrium (NE)**, i.e., a state where neither the discriminator nor the generator can improve their cost unilaterally [206]. The generator and discriminator train themselves simultaneously [1] for NE. On the contrary, when both G and D update their cost function independently without coordination, it is hard to achieve NE. Thus, GAN training becomes unstable. The NE is essential to stabilize the GANs training, which is best, explains with the help of the following example:

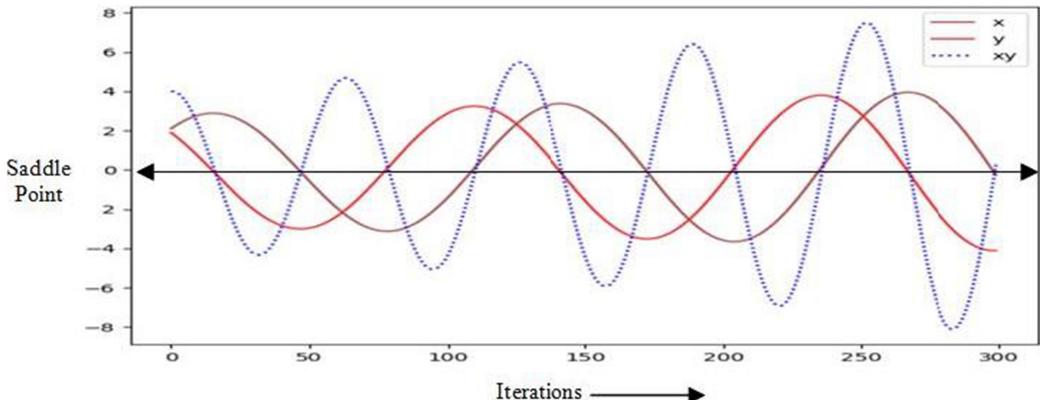


Fig. 10. With more iteration, the oscillation grows-up, which causes more and more, training instability. Image from Reference [206].

For instance, suppose that there are two players, A and B, which manage the value x and y . Player A desires to minimize the value (xy) while B wants to maximize the value function:

$$V(x, y) = xy. \quad (4)$$

By solving Equation (4), $\partial_x V(x, y) = 0$ and $\partial_y V(x, y) = 0$, we can determine that $x = y = 0$ has a saddle point (Nash equilibrium).

Let us see whether we can find the Nash equilibrium using a gradient descent optimizer. For this, we update the parameter x and y based on the gradient descent of the value function V (Equation (4)) describes through the partial differential equations system:

$$\Delta x = \alpha \frac{\partial (xy)}{\partial (x)}, \quad (5)$$

$$\Delta y = -\alpha \frac{\partial (xy)}{\partial (y)}, \quad (6)$$

where α is the learning rate, by plotting x , y , and xy against the training iterations, we realize that our solution does not converge. After every gradient update causes massive fluctuation, and instability becomes poor as shown in Figure 10.

5.1.2 Internal Covariate Shift. Internal Covariate Shift occurs when the input distribution of network activation differs from updating parameters in previous layers [207]. When the input distribution of the network changes, intermediate layers (hidden layers) try to adapt to the new distribution, these learning parameters slow down the model's training due to a change in learning rates. Due to the updated learning-rates, the model required much longer training time to counter these shifts. The longer time automatically increases the training cost, because the model reserved the resources in a higher time.

5.1.3 Mode Collapse. The **mode collapse (MC)** problem is the most crucial topic associated with GAN training, where the generator always produces identical outputs. The MC is a common cause of GAN training failure. A generator demonstrates low diversity amongst data or generates only specific types of real samples, limiting the learned GAN's usefulness in many applications of computer vision and computer graphics. The MC may be of partial or complete type. The partial type of MC produces images with small diversity, and the completer type (worst-case scenario) provides images of a single kind with no variety [208, 209].

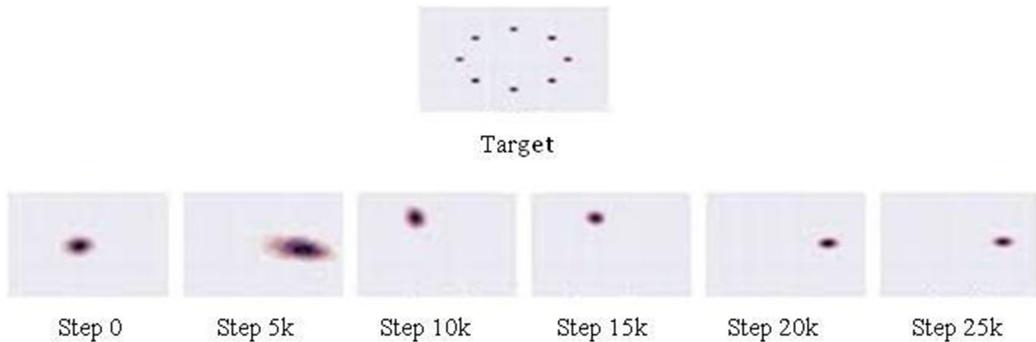


Fig. 11. Visual example of the MC problem. Image from U-GAN Reference [212].

Figure 11 shows the model collapse for a 2D toy example where the top row shows the objective distribution the model should learn. The base row shows the n of distributions the model learned during the training. As we see in Figure 11 that instead of covering all possible modes, the generator covers only one goal distribution mode. The generator then switches to a different mode as the discriminator learns to reject the generator's selected mode.

5.1.4 Vanishing Gradient. GAN is measured tough to train due to the **vanishing gradient** (VG) problem, and the generator may fail to improve in producing good quality images. The VG is due to the generator's gradients concerning weights in earlier layers of the networks becoming very small, like vanishingly small that initial layers stopped to learn [208]. The discriminator (well-trained network) rejecting the generator-generated samples with confidence due to VG [1]. Optimizing the generator may be challenging, because the discriminator does not share any information, which harms the models' learning capacity.

5.1.5 Lack of Proper Evaluation Metrics. Evaluation of GAN models is an active research area regarding unstable training. Despite the GAN grand success achieved in numerous CV applications, it is still difficult to evaluate which method(s) is better than other methods [210]. There is no standard defined function for evaluation. Most of the papers proposed that the best GAN introduced their new evaluation techniques. Thus, there are no standard consensus parameters for fair model comparison, which hurt the GAN performance.

5.2 Techniques to Improve GAN Training

GAN unstable training is one of the biggest problems due to different factors that have been discussed in previous section. In this section, we survey several solutions in detail that are crucial for successfully implementing the generative models. Supplemental material presents a comparative analysis across GAN training techniques in terms of their pros and cons.

5.2.1 Feature Matching. This technique improves the training stability of GAN. **Feature matching (FM)** [211] introduces a new cost function for the generator by substituting the discriminator's output in GAN loss function in Equation (3) to prevent over-fitting from the current discriminator. The generator generates data in the newly introduced cost function that must reflect the actual data statistics. The discriminator must discriminate the actual data's statistical data rather than explicitly maximizing the discriminator's output. The FM is a practical approach

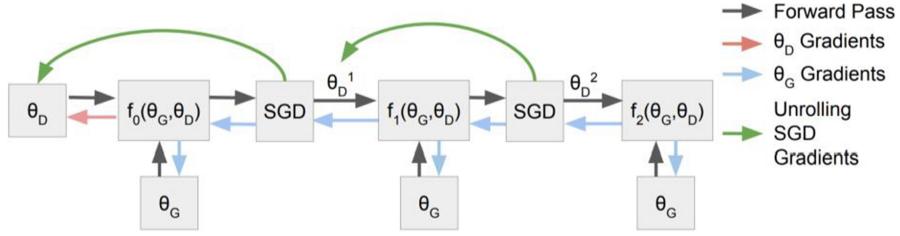


Fig. 12. Example of U-GAN computation for three unrolling steps, where the generator and the discriminator update their parameters through Equation (12) and Equation (13), respectively. Each unrolling step k uses the gradients of f_k concerning parameter (θ_D^k) defined in Equation (14). Image from Reference [212].

to unstable GAN. The new modified cost function is defined as follows:

$$\|\mathbb{E}_{x \sim p_{data}} f(x) - \mathbb{E}_{z \sim p_z(z)} f(G(z))\|_2^2, \quad (7)$$

where $f(x)$ show discriminator intermediate activation and $f(G(z))$ show features maps of generated data.

5.2.2 Unrolled GAN. As stated earlier, the model collapse is critical when GAN becomes unstable during training. The **unrolled GAN (U-GAN)** [212] approach solves the problem of model collapse. In U-GAN, the generator updated by unrolling the discriminator updates steps in contrast to standard GAN, where the discriminator was first updated by keeping the generator fixed. Then the generator is updated for the updated discriminator during training. U-GAN can noticeably reduce the mode dropping problem and improve the stability of GAN during training (see Figure 12).

The optimal parameter (θ_D^*) for D is defined as:

$$\theta_D^0 = \theta_D, \quad (8)$$

$$\theta_D^{k+1} = \theta_D^k + \eta^k \frac{df(\theta_G, \theta_D^k)}{d\theta_D^k}, \quad (9)$$

$$\theta_D^* = \lim_{k \rightarrow \infty} \theta_D^k, \quad (10)$$

where η^k is the learning rate, θ_G represents a parameter for generator, and θ_D represents discriminator parameter networks, respectively.

By unrolling the discriminator optimization for the parametric value of K steps, a new surrogate objective function for the updates of the generator network is as follows:

$$f_k(\theta_G, \theta_D) = f(\theta_G, \theta_D^k(\theta_G, \theta_D)). \quad (11)$$

As shown in Equation (11) that the updated lost function for the generator network can be controlled by adjusting the value of unrolling steps, i.e., parameter K 's value.

This new surrogate objective function (Equation (11)) used for the generator and discriminator parameters update is defined as follows:

$$\theta_G \leftarrow \theta_G - \eta \frac{df_k(\theta_G, \theta_D)}{d\theta_G}, \quad (12)$$

$$\theta_D \leftarrow \theta_D + \eta \frac{df(\theta_G, \theta_D)}{d\theta_D}. \quad (13)$$

By examining the surrogate loss $f_k(\theta_G, \theta_D)$, gradient concerning the generator parameters θ_G :

$$\frac{df_k(\theta_G, \theta_D)}{d\theta_G} = \frac{\partial f(\theta_G, \theta_D^k(\theta_G, \theta_D))}{\partial \theta_G} + \frac{\partial f(\theta_G, \theta_D^k(\theta_G, \theta_D))}{\partial \theta_D^k(\theta_G, \theta_D)} \frac{d\theta_D^k(\theta_G, \theta_D)}{d\theta_G}. \quad (14)$$

The first expression represents the GAN gradient, and the second expression reflects how the discriminator responds to changes in the generator on the left side of Equation (14). If the generator tendency to collapse to one mode, then the discriminator raises the loss for the generator. Thus, the U-GAN can put-off the mode collapse trouble for GAN model.

5.2.3 Mini-batch Discrimination. **Mini-batch discrimination (MBD)** [211] solves the problem of mode collapse during GAN training. In this technique, the discriminator processes multiple data examples in mini-batches instead of processing each data example independently to avoid the generator's mode collapse. The MBD is a multi-step process that consists of the following steps to add mini-batch discrimination to the network.

Let $f(x_i) \in \mathbb{R}^A$ denotes the feature vector of the input data x_i and $T \in \mathbb{R}^{A \times B \times C}$ represents the tensor vector. Mini-batch discrimination is a multi-step procedure defined in the form of the following steps:

Generate a matrix M with j rows through multiplying the vector $f(x_i)$ by a tensor $T \in \mathbb{R}^{A \times B \times C}$

$$M_i \in \mathbb{R}^{B \times C}. \quad (A)$$

Then, calculates the L_1 -distance between the rows of the resulting matrix M_j , through samples $i \in \{1, 2, \dots, n\}$ and then apply the following define negative exponential:

$$c_b(x_i x_j) = \exp(-\|M_i, b - M_j, b\|_{L_1}) \in \mathbb{R} \quad (B)$$

Then, the output $o(x_i)$ for a x_i is then defined as the sum of the $c_b(x_i, x_j)$ to all other samples:

$$o(x_i) = \sum_{j=1}^n (c_b(x_i, x_j)) \in \mathbb{R}. \quad (C)$$

Then, concatenate $o(x_i)$ with $f(x_i)$:

$$x_i = [o(x_i) 1, o(x_i) 2, \dots, o(x_i) B] \in \mathbb{R}^B \quad (D)$$

$$o(X) \in \mathbb{R}^{n \times B}, \quad (15)$$

where $o(X)$ in Equation (15) is the result of the MBD [211] operation for the input data sample x_i .

5.2.4 Historical Averaging. **Historical Averaging (HA)** [211] can increase the GAN training stability. This approach takes the average of the parameters ($\theta[i]$) in the past at a time (i) and adds the average value to the respective cost functions of both G and D . Historical averaging is considering during the networks parameters updates represented mathematically such as:

$$\left\| \theta - \frac{1}{t} \sum_{i=1}^t \theta[i] \right\|^2, \quad (16)$$

where the value of model parameters (θ) at the past time (i) is represented by $\theta[i]$.

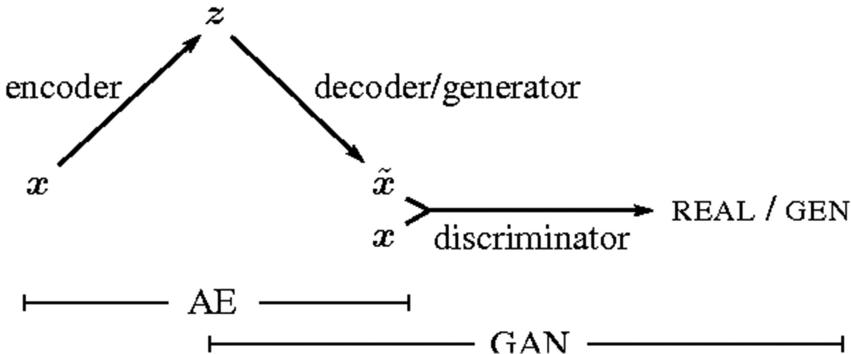


Fig. 13. The simple architecture of VAE-GAN. Image from Reference [214].

5.2.5 Two timescale Update rule. **Two timescale update rule (TTUR)** approach uses different learning levels for G and D to speed up GAN training. It creates equilibrium among G and D 's parameters to update GAN training steps [213]. In TTUR, the generator network uses a slower learning rate, and the discriminator network uses a faster learning rate to reach a Nash equilibrium status. The discriminator network trains four times the learning-rate compared to the generator network. A higher learning-rate thus eases the problem of the regularized discriminator's slow learning. They experimentally showed that different learning rates could accelerate the GAN training process.

5.2.6 Hybrid Model. Every generative model has its pros and cons, like VAE [10] suffer less from the mode collapse problem but generates blurry and low-quality images. However, GAN [1] can generate sharper samples of high-quality but suffer more from mode collapse. Different approaches have been suggested to combine VAE and GAN models into one framework called the hybrid model.

Variational Autoencoders GAN (VAE-GAN) [214] (see Figure 13) architecture was proposed to demonstrate that VAE can generate more realistic images by adding a GAN discriminator for adversarial training. The architecture of VAE-GAN shares the decoder network of a VAE with the generator network of a GAN in-line to supplement the reconstruction loss with a learned similarity metric, which is considered in a feature-space instead of being in an image-space. The VAE-GAN method distinguishes itself from GANs in that the generators of GAN are now variational autoencoder. Jointly training VAE with an adversarial loss in a single network can consistently produce sharp and realistically looking high-quality samples, because the pixelwise errors used for the training of VAE are replaced with a learned similarity metric from the GAN discriminator. After all, the models trained with learned similarity metrics generate improved quality samples than the models trained with element-wise errors. This combines multiple losses approach achieved the state-of-the-art result from the previous approaches and avoided the mode of collapse during the model's training.

Another hybrid model called **transferred deep-convolutional generative adversarial network (T-DCGAN)** [215] was proposed to solve malware data detection. The T-DCGAN model takes a Deep Autoencoder as a generator network and a Deep Convolution GAN as a discriminator network to detect zero-day attacks in malware (zero-day attacks are those attacks that are mysterious before their detection). The T-DCGAN model first generates fake malware data and then learns to distinguish fake malware data from real malware data. This junction of VAE and GAN architectures gives better outcomes. The T-DCGAN model achieves meaningful results on malware data detection, particularly in preventing the mode collapse issue during the model's training and

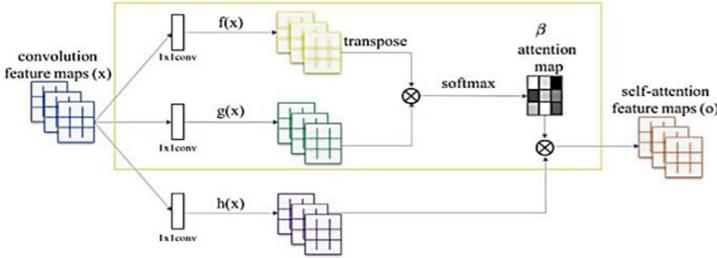


Fig. 14. $f(x)$ and $g(x)$ represent the two feature spaces that first transform the object features from the previously hidden layer to measure the attention map (β). \otimes indicates matrix multiplication. Image from Reference [216].

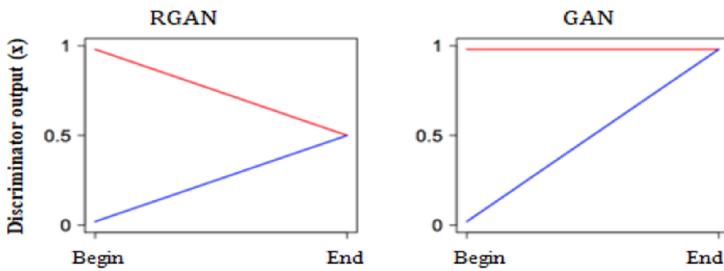


Fig. 15. R-GAN’s discriminator does better than standard GAN’s discriminator in distinguishing between real or fake samples. R-GAN’s discriminator pushes toward 0.5 rather than pushes toward 1, as it is happening in standard GAN. R-GAN has stable training than standard GAN. Image from Reference [220].

exhibiting better training stability than other state-of-the-art GAN-derived models. Similarly, IAN [66] also achieved better-customized results by combining the VAE and GAN models.

5.2.7 Self-attention GAN. In **Self-attention GAN (SA-GAN)**, the information from a broader feature space across image regions can create samples with a fine-detailed resolution instead of information spread in local neighbourhoods [216] (see Figure 14). SA-GAN can generate multi classes images by coordinating the fine details of every location with distant portions. Similarly, machine interpretation models [217] accomplish the best results by completely utilizing an SA-mechanism. The image transformer model [218] was introduced for image generation with an SA-mechanism into an autoregressive model. In non-local neural networks [219] formalized, the SA-mechanism was a non-nearby activity to demonstrate the video sequence’s spatial-transient conditions.

5.2.8 Relativistic GAN. Relativistic GAN (R-GAN) claims that the generator training should not only enhance the possibility that the synthesis sample is actual but also reduce the likelihood that the actual sample is actual [220]. The discriminator in R-GAN figures out how to predict “if an image is more real than the second one” instead of “if an image is actual or synthesized.” Experimental results show that R-GAN has better training stability than vanilla GAN (see Figure 15). In standard formulation, the new loss function for R-GAN is stated as

$$\min_{D} \mathbb{E}_{\substack{x_r \sim p_r \\ x_g \sim p_g}} [\log(\text{sigmoid}(C(x_r) - C(x_g)))], \quad (17)$$

$$\min_{G} \mathbb{E}_{\substack{x_r \sim p_r \\ x_g \sim p_g}} [\log(\text{sigmoid}(C(x_g) - C(x_r)))], \quad (18)$$

where $C(x)$ is a function that gives a score to the input image (evaluates how much x is realistic), and sigmoid translates the score into a probability between zero to 1.

5.2.9 One-sided Label Smoothing. This technique provides smooth label values to the discriminator, which significantly improves the training. GAN's vulnerability was reduced by feeding the labels with softening value, i.e., slightly less than 1, such as 0.9 for the real sample, and slightly higher than 0, such as 0.1 for fake examples instead of 1 and 0 [221]. Label smoothing is highly useful for the training of high-quality networks on relatively modest-sized training datasets.

5.2.10 Sampling GAN. Sampling GAN uses a spherical linear interpolation method for data sampling instead of the linear interpolation method [222]. In SGAN, each section bounded by two data points interpolated independently. They experimentally showed that the spherical linear interpolation method performs better than linear interpolation methods.

5.2.11 Proper Optimizer. The use of appropriate optimization algorithms such as Root Mean Square Propagation [223], Stochastic Gradient Descent [224], Adaptive Gradient Algorithm [225], and **Adaptive Moment Estimation (ADAM)** [226] can dramatically improve the training stability of GAN. Experiments with different optimizers show that ADAM is the most popular optimizer among all other optimizers for severe dimensional problems and improves convergence speed and GAN training stability.

5.2.12 Normalization. GANs are computationally expensive due to their more considerable training time. The training time of GAN can be reduced with the help of normalizing operations. Different normalization techniques are proposed to minimize training time, which improves the stability of GAN during training. Supplementary Material reviews various normalization techniques that can stabilize GAN training.

5.2.13 Add Instance Noise. Adding the instance noise to both the real and generated data can increase GAN's training stability [227, 228]. The level of additive independent Gaussian noise can be toughening throughout the training process. The noise allows us to optimize the discriminator until convergence is reached in each iteration carefully. This is beneficial in fixing the instability and vanishing gradient issue of GAN training.

5.2.14 Train with Labels. Adding the label as part of the latent space improves the GAN training. The conditioning model can manage the generation of samples with its conditional variables; applied to both G and D . Conditional GAN [15] can be constructed by merely feeding the extra information in the model that controls data generation is impossible to control with vanilla GAN. Conditional GAN is used in many computer vision tasks such as Age-CGAN [49] introduced for the generation of high-quality images by considering the age condition, AC-GAN [56] for conditional image synthesis, and CGAN-based cartoon image generation [104] from the sketch, and so on.

5.2.15 Alternative Cost Functions. Alternative cost functions prevent mode collapse and vanishing gradient problems. Least-Square GAN (LS-GAN) [229] proposed a new cost function where sigmoid cross-entropy loss is replaced with a least-square loss to overcome vanishing gradient trouble. A newly introduced least square loss in LS-GAN classifies the actual and fake samples and brings the fake data nearer to the real data for better architecture and real-world data generation. The updated GAN loss from Equation (3) for LS-GAN loss is defined as follows:

$$\min_D V_{LSGAN}(D) = \frac{1}{2} \mathbb{E}_{x \sim p_{data}(x)} [(D(x) - b)^2] + \frac{1}{2} \mathbb{E}_{z \sim p_z(z)} [(D(G(z)) - a)^2], \quad (19)$$

$$\min_G V_{LSGAN}(G) = \frac{1}{2} \mathbb{E}_{z \sim p_z(z)} [(D(G(z)) - c)^2], \quad (20)$$

where a , b , and c are the correspondence with the labels for the fake samples, the real samples, and the value that G desires D to consider for the produced samples. The discriminator to get actual values rather than getting probabilities values for real or generated values, respectively.

5.2.16 Gradient Penalty. WGAN Gradient Penalty (WGAN-GP) [21] suggests that adding a **gradient penalty (GP)** term in place of the weight clipping would improve the model performance and the stability of GAN during training. In short, GP's use dramatically enhances the training stability and reduces the mode collapse of the networks. WGAN-GP has better convergence power, improved training speed, and sample quality than other robust GAN models by pushing the discriminator network to learn smoother decision boundaries.

5.2.17 Representative Features-based GAN. A **Representative features-based GAN (RF-GAN)** [230] model was proposed using pre-trained encoder features to regularize discriminator training to mitigate the mode collapse problem. Representative features are also used along-with discriminative features to train the discriminator of the standard GAN. This extra training feature of the discriminator helps break the tradeoff between samples' quality and variety. RF-GAN-based training technique helps avoid the mode collapse via pre-trained encoder features.

5.2.18 Cycle-consistency Loss. **Cycle-consistent GANs (CC-GAN)** [108] is based on the design that when we transfer an image from a domain (A) to domain (B) (i.e., A→B), and then transfer back from a domain (B) to domain (A) (i.e., B→A), the obtained result should be similar to the actual input image in the presence of non-paired examples. During training, the model minimizes the distance between original and reconstructed images. The same thing is also done in the reverse direction, i.e., for the domain (B). The CC-GAN presents a cycle-consistency loss by coupling two different pairs of GANs together. The first pair of GANs translates the image into another domain, and the second pair of GANs does the inverse without paired training data. It is proved that cycle-consistency loss with adversarial loss can protect the actual image after a cycle of forwarding and backward translations. The CC-GAN architecture prevents generators from extreme hallucinations and collapsing mode by enforcing cycle-consistency mechanisms. So, the cycle consistency loss plays an important role in stabilizing GAN training via escaping the mode collapse issue.

5.3 Experiments

In this section, experiments are conducted to assess the GAN stabilization training techniques on the practical and simple CIFAR-10 dataset [249]. Here, the effects on the GAN training stability of different stabilization training techniques are compared. The dataset, evaluation metrics, experiment implementation details, experimental results, and analysis and comparisons of quantitative results are described next. Here, we try to execute the training techniques in a set of experiments successfully.

5.3.1 CIFAR-10. We used a benchmark CIFAR-10 dataset for all the experiments, which has 50,000 training and 10,000 test images. All images are 32×32 pixels in size, with ten different classes, including aircraft, vehicles, cats, ships, dogs, birds, frogs, horses, deer, and trucks. Each class consists of 6,000 images. The reason for selecting the CIFAR-10 is its extensive utilization in the testing of numerous machine learning algorithms.

5.3.2 Evaluation Metrics. The different generative models' performance is mostly calculated concerning the diversity and quality of the generated samples. The quality and diversity of the generated samples are quantitatively calculated by using different evaluation criteria metrics. Here,

Table 8. Quantitative Scores of Different Training Techniques

Training technique	Evaluation metrics.	
	IS↑	FID↓
Train with labels (C-GAN)	6.99	36.4
Deep regret analytic GAN (DRA-GAN)	6.29	56.3
Spectral normalization (SN-GAN)	6.27	55.9
Batch normalization (BN-GAN)	6.20	55.1
Two-time scale update rule	6.14	54.1
Feature matching (FM-GAN)	5.95	62.9
Relativistic GAN (R-GAN)	5.90	62.3
Alternative loss functions	5.87	62.1
Representative features (RF-GAN)	5.74	62.9
Historical averaging (HA-GAN)	5.73	63.7
Cycle-consistency loss (Cycle-GAN)	5.70	63.2
One-sided label smoothing	5.69	63.1
Proper optimizer (Adam)	5.62	61.1
Mini-batch discrimination	5.47	64.8
Self-attention (SA-GAN)	5.41	64.5

the effect of several training techniques on training stability is quantitatively assessed through **Inception-Score (IS)** [211] and **Fréchet Inception Distance (FID)** [213], which are the most commonly accepted GAN evaluation metrics to measure the diversity and quality of the generated samples, because it significantly correlates with human findings of the generated samples by the model. A higher IS value shows good performance, while a lower IS value shows the model's bad performance. A lower FID value shows good performance, while a higher FID value shows the bad performance of the model. A better quantitative score shows that the model can generate diverse samples of good quality over several trainings, indicating improved stability.

5.3.3 Implementation Details. All the methods are trained and tested with Nvidia GTX 1080Ti GPU to generate 32×32 images CIFAR-10 dataset. For the sake of fair comparison, we trained all the models 10 times under the same network architecture and training hyper-parameters for 200K iterations with Adam optimizer [226].

5.3.4 Experimental Results. The quantitative results in terms of IS and FID demonstrate the effectiveness of the different GAN stabilization training techniques' training stability. We compared the IS and FID of different training techniques for 200K epochs on the CIFAR-10 dataset. The average evaluated IS and FID scores of different training techniques are provided in Table 8.

5.3.5 Analysis and Comparisons. Before giving a short analysis of the experimental results, we must keep in mind that direct comparisons of these techniques are not, possible because they are trained and tested on different datasets under various experimental settings. Thus, for a fair comparison, we used a benchmark of the CIFAR-10 dataset and a benchmark of IS and FID evaluation metrics. Table 8 shows the quantitative scores of several training techniques on the CIFAR-10 dataset. It is observed from quantitative scores that the reviewed training techniques of GAN have considerable impacts. As we can see in Table 8 that highest IS (6.99) indicate that training with labels training techniques is the most effective GAN stabilization training technique. The lowest FID (36.4) also validates that training with labels training techniques is the most effective GAN stabilization training technique. The experimental results show that training with labels training techniques is more stable than all other methods on the CIFAR-10 dataset discussed in this

survey. However, several GAN-derived models' general performance still depends on the application context and experimental settings. In the absence of a robust and consistent evaluation method, it is challenging to perform a reasonable evaluation among GAN training techniques. As literature review shows that proposed training methods used different architectures and loss functions under different parameter settings. Therefore, it is not necessarily true that a GAN-derived model's bad presentation under a particular experimental setting means bad performance under other experimental settings. It is challenging to compare different training methods under the same experimental setting following the same training procedure. It results in some methods having a downfall in their performance.

6 CONCLUSION AND FUTURE RESEARCH DIRECTIONS

This study presented a GAN survey, GAN-variants, and a detailed analysis of different GAN applications in various research areas. Despite all these, this survey's core idea is to discuss the GAN model training obstacles and their potential solutions to advance GAN training. The above discussion shows that GAN has the power to facilitate several new practical applications in many other domains, like image, audio, and video, in the future. Despite GAN's significant success, the architecture of GAN still faces problems due to unstable training. Thus, we discussed several training techniques had recommended by different researchers to stabilize training and have fixed some previous limitations for the generation of highly realistic-looking data. Irrespective of the significant progress of GAN models in recent years, several issues remain to deal with for future research, such as:

- **Discrete data:** GAN can generate continuous value data but cannot generate discrete value data (words, characters, bytes) directly on which **natural language processing (NLP)** applications heavily based; thus, GAN capability for NLP applications becomes limited. Different researchers proposed different methods such as WGAN-GP [21] model discrete data with a continuous generator, Boundary-Seeking GAN (BSGAN) [231] introduced a new way for the training of GAN with discrete data, Maximum-Likelihood GAN [232] purposed a new objective function for the discriminator network. Adversarially Regularized AE [233] proposed a new method for training a discrete structure. This appealing area deserves more effort to be done.
- **Training instability:** A stable training is essential to reach Nash equilibrium [206] for the generator to capture the real distribution of actual examples, but it is still difficult for both G and D to find a saddle point. Early attempts have made toward this direction of research, such as W-GAN [20], WGAN-GP [21], WGAN Lipschitz Penalty [234], FM [211], MBD [211], One-sided Label Smoothing [221], HA [211], TTUR [213], and **Spectral Normalization GAN (SN-GAN)** [235] for more stable training. Training instability is another crucial future problem, since more solutions are needed to stabilize GAN training.
- **Model evaluation:** GAN model evaluation is one of the directions that need to be discussed in the future, because it is still difficult to evaluate which method (s) is better than other methods. Despite the early efforts toward that are broadly used at current such as IS [211], Mode Score [234], Kernel Inception Distance [236], FID [213], Multi-Scale Structural Similarity [237], Classifier Two-Sample Tests [238], and Maximum Mean Discrepancy [239], there are no standard consensus parameters for fair model-to-model comparison. Thus, the development of better and universal quantitative evaluation metrics for this scenario still requires future research.
- **Model collapse:** As discussed earlier that GAN does suffer mode collapse, i.e., the situation in which the generator generates multiple samples with similar properties or minimal diversity amongst generated samples. Hence, to increase the diversity of the samples produced,

different solutions have also been proposed, such as W-GAN [20], MBD [211], **Energy-Based GAN (EB-GAN)** [19], U-GAN [212], DRA-GAN [36], Adaptive GAN [240], Mode Regularized GAN [241], Multi-Agent Diverse GAN [242] show that how these methods solve mode collapse and increase the diversity of the generated samples with higher visual quality. But the development of a better training algorithm for this scenario still requires future research.

—**Others:** GANs also face other research issues such as disease prediction, Nash equilibrium, and vanishing gradient.

We hope this study will help the researchers to get a comprehensive overview of Generative Adversarial Networks.

REFERENCES

- [1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Neural Information Processing Systems*, 2672–2680.
- [2] Jie Gui, Zhenan Sun, Yonggang Wen, Dacheng Tao, and Jieping Ye. 2020. A review on generative adversarial networks: Algorithms, theory, and applications. arXiv:2001.06937. Retrieved from <https://arxiv.org/abs/2001.06937/>.
- [3] Zhengwei Wang, Qi She, and Tomas E. Ward. 2019. Generative adversarial networks: A survey and taxonomy. arXiv:1906.01529. Retrieved from <https://arxiv.org/abs/1906.01529>.
- [4] Zhaoqing Pan, Weijie Yu, Xiaokai Yi, Asifullah Khan, Feng Yuan, and Yuhui Zheng. 2019. Recent progress on generative adversarial networks (gans): A survey. *IEEE Access* 7 (2019), 36322–36333.
- [5] Yongjun Hong, Uiwon Hwang, Jaeyoon Yoo and Sungroh Yoon. 2019. How generative adversarial nets and its variants work: An overview. *ACM Comput. Surv.* 52, 1 (2019), 1–43.
- [6] Saifuddin Hitawala. 2018. Comparative study on generative adversarial networks. arXiv:1801.04271. Retrieved from <https://arxiv.org/abs/1801.04271>.
- [7] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. 2018. Generative adversarial networks: An overview. *IEEE Sign. Process. Mag.* 35, 1 (2018), 53–65.
- [8] Mahima Vuppuluri and Abhishek Dash. 2017. Survey on generative adversarial networks. *International Journal of Engineering Research in Computer Science and Engineering*. 11, 4 (IJERCSE'17).
- [9] Geoffrey E. Hinton and Richard S. Zemel. 1993. Autoencoders, minimum description length, and helmholtz free energy. In *Proceedings of the International Conference on Neural Information Processing Systems*.
- [10] Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representation*. 1–14.
- [11] Diederik P. Kingma and Prafulla Dhariwal. 2018. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*. 10 236–10 245.
- [12] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. 2019. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems*.
- [13] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*. 1747–1756.
- [14] Alexander Kolesnikov and Christoph H. Lampert. 2017. Pixelcnn models with auxiliary variables for natural image modeling. *Int. Conf. Mach. Learn* 70 (2017), 1905–1914.
- [15] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. arXiv:1411.1784. Retrieved from <https://arxiv.org/abs/1411.1784>.
- [16] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434. Retrieved from <https://arxiv.org/abs/1511.06434>.
- [17] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. 2015. Deep generative image models using a laplacian pyramid of adversarial networks. In *Neural Information Processing Systems* 1486–1494.
- [18] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever‡, and Pieter Abbeel. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the Neural Information Processing Systems*, 2172–2180.
- [19] Junbo Zhao, Michael Mathieu, and Yann LeCun. 2016. Energy-based generative adversarial network. arXiv:1609.03126. Retrieved from <https://arxiv.org/abs/1609.03126>.
- [20] Martin Arjovsky, Soumith Chintala, and Leon Bottou. 2017. Wasserstein generative adversarial networks. In *Proceedings of the International Conference on Machine Learning*. 214–223.
- [21] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. Improved training of wasserstein gans. In *Neural Information Processing Systems* 5767–5777, 2017.

- [22] David Berthelot, Thomas Schumm, and Luke Met. 2017. Began: Boundary equilibrium generative adversarial networks. arXiv:1703.10717. Retrieved from <https://arxiv.org/abs/1703.10717>.
- [23] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2018. Progressive growing of gans for improved quality, stability, and variation. arXiv:1710.10196. Retrieved from <https://arxiv.org/abs/1710.10196>.
- [24] Andrew Brock, Jeff Donahue, and Karen Simonyan. 2019. Large scale gan training for high fidelity natural image synthesis. In *Proceedings of the International Conference on Learning Representations*, 2019. 1–35.
- [25] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4401–4410.
- [26] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* 115, 3 (2015), 211–252.
- [27] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1125–1134.
- [28] Yuchen Tian. 2017. Zi2zi: Master chinese calligraphy with conditional adversarial networks.
- [29] Bo Chang, Qiong Zhang, Shenyi Pan, and Lili Meng. 2018. Generating handwritten chinese characters using cycle-gan. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV'18)*. 199–207.
- [30] Xianming Lin, Jie Li, Hualin Zeng, and Rongrong Ji. 2018. Font generation based on least-squares conditional generative adversarial nets. *Multimedia Tools Appl.* (2018).
- [31] Hideaki Hayashia, Kohtaro Abea, and Seiichi Uchidaa. 2019. Glyphgan: Style-consistent font generation based on generative adversarial networks. arXiv:1905.12502. Retrieved from <https://arxiv.org/abs/1905.12502>.
- [32] Jin Liu, Chenkai Gu, Jin Wang, Geumran Youn, and Jeong-Uk Kim. 2019. Multi-scale multi-class conditional generative adversarial network for handwritten character generation. *J. Supercomput.* 4 (2019), 1922–1940.
- [33] Bo Ji and Tianyi Chen. 2019. Generative adversarial network for handwritten text. arXiv:1907.11845. Retrieved from <https://arxiv.org/abs/1907.11845>.
- [34] Md. Fahim Sikder. 2019. Bangla handwritten digit recognition and generation. In *Proceedings of the International Joint Conference on Computational Intelligence*. 547–556.
- [35] Yanghua Jin, Jiakai Zhang, Minjun Li, Yingtao Tian, Huachun Zhu, and Zhihao Fang. 2017. Towards the automatic anime characters creation with generative adversarial networks. arXiv:1708.05509. Retrieved from <https://arxiv.org/abs/1708.05509>.
- [36] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. 2017. How to train your dragan. arXiv:1705.07215. Retrieved from <https://arxiv.org/abs/1705.07215>.
- [37] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1874–1883.
- [38] Frontwing. 2017 (web services hosting Japanese games). Retrieved from <http://www.getchu.com/soft.phtml?id=933144>.
- [39] Koichi Hamada, Kentaro Tachibana, Tianqi Li, Hiroto Honda, and Yusuke Uchida. 2018. Full-body high-resolution anime generation with progressive structure-conditional generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV'18)*.
- [40] Huihai Wu, Shuai Zheng, Junge Zhang, and Kaiqi Huang. 2017. Gpgan: Towards realistic high-resolution image blending. arXiv:1703.07195. Retrieved from <https://arxiv.org/abs/1703.07195>.
- [41] Robert T. Frankot and Rama Chellappa. 1988. A method for enforcing integrability in shape from shading algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.* (1988).
- [42] Patrick Perez, Michel Gangnet, and Andrew Blake. 2003. Poisson image editing. *ACM Trans. Graph.* (2003).
- [43] Bor-Chun Chen and Andrew Kae. 2019. Toward realistic image compositing with adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8407–8416.
- [44] B. Dolhansky and C. C. Ferrer. 2018. Eye in-painting with exemplar generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7902–7911.
- [45] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. 2018. Generative image inpainting with contextual attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5505–5514.
- [46] U. Demir and G. Unal. 2018. Patch-based image in-painting with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*.
- [47] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. Globally and locally consistent image completion. *ACM Trans. Graph.* 36, 4 (2017), 107.
- [48] Chuan Li and Michael Wand. 2016. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 702–716.
- [49] Grigory Antipov, Moez Baccouche, and Jean-Luc Dugelay. 2017. Face aging with conditional generative adversarial networks. arXiv:1702.01983. Retrieved from <https://arxiv.org/abs/1702.01983>.

- [50] Zhifei Zhang, Yang Song, and Hairong Qi. 2017. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*. 5810–5818.
- [51] Zongwei Wang, Xu Tang, Weixin Luo, and Shenghua Gao. 2018. Face aging with identity-preserved conditional generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- [52] Peipei Li, Yibo Hu, Ran He, and Zhenan Sun. 2018. Global and local consistent wavelet-domain age synthesis. arXiv:1809.07764. Retrieved from <https://arxiv.org/abs/1809.07764>.
- [53] Yunfan Liu, Qi Li1, and Zhenan Sun. 2019. Attribute aware face aging with wavelet-based generative adversarial networks. arXiv:1809.06647. Retrieved from <https://arxiv.org/abs/1809.06647>.
- [54] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. 2018. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*. 1316–1324.
- [55] Han Zhang, Tao Xu, Hongsheng Li, Shaotong Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. 2016. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. arXiv:1612.03242. Retrieved from <https://arxiv.org/abs/1612.03242>.
- [56] Augustus Odena, Christopher Olah, and Jonathon Shlens. 2017. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the International Conference on Machine Learning*, 2017. 2642–265.
- [57] Ayushman Dash, John Gamboa, and Sheraz Ahmed. 2017. Tac-gan-text conditioned auxiliary classifier generative adversarial network. arXiv:1703.06412. Retrieved from <https://arxiv.org/abs/1703.06412>.
- [58] Hao Dong, Simiao Yu, Chao Wu, and Yike Guo. 2017. Semantic image synthesis via adversarial learning. In *Proceedings of the IEEE International Conference on Computer Vision*. 5706–5714.
- [59] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. 2018. Pose guided person image generation. arXiv:1705.09368. Retrieved from <https://arxiv.org/abs/1705.09368>.
- [60] Aliaksandr Siarohin, Stephane Lathuiliere, Enver Sangineto, and Nicu Sebe. 2019. Appearance and pose conditioned human image generation using deformable gans. arXiv:1905.00007. Retrieved from <https://arxiv.org/abs/1905.00007>.
- [61] Denis Volkonski, Ivan Nazarov, Boris Borisenko, and Evgeny Burnaev. 2017. Steganographic generative adversarial networks. arXiv:1703.05502. Retrieved from <https://arxiv.org/abs/1703.05502>.
- [62] Haichao Shia, Jing Dongc, Wei Wangc, Yinlong Qianc, and Xiaoyu Zhang. 2017. Ssgan: Secure steganography based on generative adversarial networks. arXiv:1707.01613. Retrieved from <https://arxiv.org/abs/1707.01613>.
- [63] Kevin A. Zhang, Alfredo Cuesta-Infante, Lei Xu, and Kalyan Veeramachaneni. 2019. Steganogan: High capacity image steganography with gans. arXiv:1901.03892. Retrieved from <https://arxiv.org/abs/1901.03892>.
- [64] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. 2016. Generative visual manipulation on the natural image manifold. In *Proceedings of the European Conference on Computer Vision*. 597–613.
- [65] Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. 2018. Text-adaptive generative adversarial networks: Manipulating images with natural language. In *Proceedings of the International Conference on Neural Information Processing*, 2018.
- [66] Andrew Brock, Theodore Lim, J. M. Ritchie, and Nick Weston. 2016. Neural photo editing with introspective adversarial networks. arXiv:1609.07093. Retrieved from <https://arxiv.org/abs/1609.07093>.
- [67] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. 2019. Attgan: Facial attribute editing by only changing what you want. *IEEE Trans. Image Process.*
- [68] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum1, William T. Freeman, and Antonio Torralba. 2019. Gan dissection: Visualizing and understanding generative adversarial networks. In *Proceedings of the International Conference on Learning Representations*. 118–129.
- [69] Junting Pana, Cristian Canton-Ferrerb, Kevin McGuinnessc, Noel E. O’Connord, Jordi Torresd, Elisa Sayrola, and Xavier Giro-i-Nieto. 2017. Salgan: Visual saliency prediction with generative adversarial networks. arXiv:1701.01081. Retrieved from <https://arxiv.org/abs/1701.01081>.
- [70] Chao Zhang, Fei Yang, Guoping Qiu, and Qian Zhang. 2019. Salient object detection with capsule-based conditional generative adversarial network. In *Proceedings of the IEEE International Conference on Image Processing*, 2019.
- [71] Prerana Mukherjee, Manoj Sharma, Megh Makwana, Ajay Pratap Singh, Avinash Upadhyay, Akkshita Trivedi, Brejesh Lall, and Santanu Chaudhury. 2019. Dsal-gan: Denoising based saliency prediction with generative adversarial networks. arXiv:1904.01215. Retrieved from <https://arxiv.org/abs/1904.01215>.
- [72] Kiana Ehsani, Roorzbeh Mottaghi, and Ali Farhadi. 2018. Segan: Segmenting and generating the invisible. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6144–6153.
- [73] Jianan Li, Xiaodan Liang, Yunchao Wei, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. 2017. Perceptual generative adversarial networks for small object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1222–1230.
- [74] Yancheng Bai, Yongqiang Zhang, Mingli Ding, and Bernard Ghanem. 2018. Sod-mtgan: Small object detection via a multi-task generative adversarial network. In *Proceedings of the European Conference on Computer Vision*. 206–221.
- [75] Charan D. Prakash and Lina J. Karam. 2019. It gan do better: Gan based detection of objects on images with varying quality. arXiv:1912.01707. Retrieved from <https://arxiv.org/abs/1912.01707>.

- [76] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T. Freeman, and Joshua B. Tenenbaum. 2016. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*. 82–90.
- [77] Matheus Gadelha, Subhransu Maji, and Rui Wang. 2016. “3d shape induction from 2d views of multiple objects. arXiv:1612.05872. Retrieved from <https://arxiv.org/abs/1612.05872>.
- [78] Cihang Ongun and Alptekin Temizel. 2018. Paired 3d model generation with conditional generative adversarial networks. In *Proceedings of the 15th European Conference on Computer Vision (ECCV’18)*.
- [79] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 2019. 3d point cloud generative adversarial network-based on tree-structured graph convolutions. In *Proceedings of the Conference on Computer Vision*.
- [80] Pranav Gandhi, Adarsh Shaw, Emil Eji, and Steffina Muthukumar. 2019. Generating 3d models using 3d generative adversarial network. *Int. Res. J. Eng. Technol.* (2019).
- [81] Yuan Xue, Tao Xu, Han Zhang, L. Rodney Long, and Xiaolei Huang. 2018. Segan: Adversarial network with multi-scale l_1 - loss for medical image segmentation. *Neuro-informatics* 16 (2018), 383–392.
- [82] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. 2017. Generating multi-label discrete patient records using generative adversarial networks. arXiv:1703.06490. Retrieved from <https://arxiv.org/abs/1703.06490>.
- [83] Mostapha Benhenda. 2017. Chemgan challenge for drug discovery: Can ai reproduce natural chemical diversity? arXiv:1708.08227. Retrieved from <https://arxiv.org/abs/1708.08227>.
- [84] Anvita Gupta, and James Zou. 2018. Feedback gan (fbgan) for dna: A novel feedback-loop architecture for optimizing protein functions. arXiv:1804.01694. Retrieved from <https://arxiv.org/abs/1804.01694>.
- [85] Ghislain St-Yves and Thomas Naselaris. 2018. Generative adversarial networks conditioned on brain activity reconstruct seen images. in *Proceedings of the Conference on Systems, Man, and Cybernetics*. 1054–1061.
- [86] Jyh-Jing Hwang, Sergei Azernikov, Alexei A. Efros, and Stella X. Yu. 2018. Learning beyond human expertise with generative models for dental restorations. arXiv:1804.00064. Retrieved from <https://arxiv.org/abs/1804.00064>.
- [87] Bing Tian, Yong Zhang, Xinhuan Chen, Chunxiao Xing, and Chao Li. 2019. Drgan: A gan based framework for doctor recommendation in chinese on-line qa communities. In *Proceedings of the International Conference on Database Systems for Advanced Applications*. 444–447.
- [88] Karim Armanious, Chenming Yang, Marc Fischer, Thomas Küstner, Konstantin Nikolaou, Sergios Gatidis, and Bin Yang. 2020. Medgan: Medical image translation using gan. *Computerized Medical Imaging and Graphics* (2020).
- [89] Tingting Li, Ruihe Qian, Chao Dong, Si Liu, Qiong Yan, Wenwu Zhu, and Liang Lin. 2018. Beautygan: Instance-level facial makeup transfer with the deep generative adversarial network. In *Proceedings of the ACM Multimedia Conference on Multimedia Conference*. 645–653.
- [90] Huiwen Chang, Jingwan Lu, Fisher Yu, and Adam Finkelstein. 2018. Pairedcyclegan: Asymmetric style transfer for applying and removing makeup. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 40–48.
- [91] Honglun Zhang, Jidong Tian, Wenqing Chen, Hao He, and Yaohui Jin. 2019. Disentangled makeup transfer with generative adversarial network. arXiv:1907.01144. Retrieved from <https://arxiv.org/abs/1907.01144>.
- [92] Xuanyi Dong, Yan Yan, Wanli Ouyang, and Yi Yang. 2018. Style aggregated network for facial landmark detection. arXiv:1803.04108. Retrieved from <https://arxiv.org/abs/1803.04108>.
- [93] Xin Yang, Yuezun Li, Honggang Qi, and Siwei Lyu. 2019. Exposing gan synthesized faces using landmark locations. arXiv:1904.00167. Retrieved from <https://arxiv.org/abs/1904.00167>.
- [94] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, and Andrew Cunningham. 2017. Photorealistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4681–4690.
- [95] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. 2018. Esgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision*. 63–79.
- [96] Jingwei Guan, Cheng Pan, Songnan Li, and Dahai Yu. 2019. Srdgan: Learning the noise prior for super-resolution with dual generative adversarial networks. arXiv:1903.11821. Retrieved from <https://arxiv.org/abs/1903.11821>.
- [97] Zihan Ding, Xiao-Yang Liu, Miao Yin, W. Liu, and Linghe Kong. 2019. Tgan: Deep tensor generative adversarial nets for large image generation. arXiv:1901.09953. Retrieved from <https://arxiv.org/abs/1901.09953>.
- [98] Chuan Li and Michael Wand. 2016. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *Proceedings of the European Conference on Computer Vision*. 702–716.
- [99] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2015. Texture synthesis using cnn. In *Advances in Neural Information Processing Systems*. 2414–2423.
- [100] Nikolay Jetchev, Urs Bergmann, and Roland Vollgraf. 2016. Texture synthesis with spatial generative adversarial networks. arXiv:1611.08207. Retrieved from <https://arxiv.org/abs/1611.08207>.

- [101] Urs Bergmann, Nikolay Jetchev, and Roland Vollgraf. 2017. Learning texture manifolds with the periodic spatial gan. In *Proceedings of the International Conference on Machine Learning*. 469–477.
- [102] Wenqi Xian, Patsorn Sangkloy, Varun Agrawal, Amit Raj, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. 2017. Texturegan: Controlling deep image synthesis with texture patches. arXiv:1706.02823. Retrieved from <https://arxiv.org/abs/1706.02823>.
- [103] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. 2017. Scribbler: Controlling deep image synthesis with sketch and color. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- [104] Yifan Liu, Zengchang Qin, Zhenbo Luo, and Hua Wang. 2017. Autopainter: Cartoon image generation from sketch by using the conditional generative adversarial networks. arXiv:1705.01908. Retrieved from <https://arxiv.org/abs/1705.01908>.
- [105] Wengling Chen and James Hays. 2018. Sketchyan: Towards diverse and realistic sketch to image synthesis. arXiv:1801.02753. Retrieved from <https://arxiv.org/abs/1801.02753>.
- [106] Jun Yu, Xingxin Xu, Fei Gao, Shengjie Shi, Meng Wang, Dacheng Tao, and Qingming Huang. 2020. Towards realistic face photo-sketch synthesis via composition aided gan. arXiv:1712.00899. Retrieved from <https://arxiv.org/abs/1712.00899>.
- [107] Chaoyue Wang, Chang Xu, Chaohui Wang, and Dacheng Tao. 2018. Perceptual adversarial networks for image-to-image transformation. *IEEE Trans. Image Process.* 27, 8 (2018), 4066–4079.
- [108] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the International Conference on Computer Vision*, 2017. 1125–1134.
- [109] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. 2017. Learning to discover cross-domain relations with generative adversarial networks. arXiv:1703.05192. Retrieved from <https://arxiv.org/abs/1703.05192>.
- [110] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. 2017. Dualgan: Unsupervised dual learning for image-to-image translation. arXiv:1704.02510. Retrieved from <https://arxiv.org/abs/1704.02510>.
- [111] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. 2018. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8789–8797.
- [112] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. 2017. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*. 700–708.
- [113] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. 2018. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV'18)*. 172–189.
- [114] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. 2018. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European Conference on Computer Vision*, 35–51.
- [115] Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Singh, and Ming-Hsuan Yang. 2019. “Drit++: Diverse image-to-image translation via disentangled representations. arXiv:1905.01270. Retrieved from <https://arxiv.org/abs/1905.01270>.
- [116] Luan Tran, Xi Yin, and Xiaoming Liu. 2017. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1415–1424.
- [117] Rui Huang, Shu Zhang, Tianyu Li, and Ran He. 2017. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *Proceedings of the International Conference on Computer Vision*. 2439–2448.
- [118] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. 2017. Towards large-pose face frontalization in the wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'17)*. 3990–3999.
- [119] Volker Blan and Thomas Vetter. 1999. A morphable model for the synthesize of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, 187–194, 1999.
- [120] Weiwei Zhuang, Liang Chen, Chaoqun Hong, Yuxin Liang, and Keshou Wu. 2019. Ftgan: Face transformation with key points alignment for pose-invariant face recognition. *Electronics (Basel)*, 8, 7 (2019), 807.
- [121] Deniz Engin, Anıl Genç, and Hazım Kemal Ekenel. 2018. Cycle-dehaze: Enhanced cycledgan for single image dehazing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 825–833.
- [122] Xitong Yang, Zheng Xu, and Jiebo Luo. 2018. Towards perceptual image dehazing by physics-based disentanglement & adversarial training. In *Proceedings of the Conference on Artificial Intelligence*. 7485–7492.
- [123] Wei Liu, Xianxu Hou, Jiang Duan, and Guoping Qiu. 2019. End-to-end single image fog removal using enhanced cycle consistent adversarial networks. arXiv:1902.01374. Retrieved from <https://arxiv.org/abs/1902.01374>.
- [124] Sebastian Lutz, Konstantinos Amplianitis, and Aljosa Smolic. 2018. Alphagan: Generative adversarial networks for natural image matting. arXiv:1807.10088. Retrieved from <https://arxiv.org/abs/1807.10088>.
- [125] Qingjie Liu, Huanyu Zhou, Qizhi Xu, Xiangyu Liu, and Yunhong Wang. 2018. Psgan: A generative adversarial network for remote sensing image pan-sharpening. In *Proceedings of the IEEE International Conference on Image Processing*. 873–877.

- [126] Fang Liu, Licheng Jiao, and Xu Tang. 2019. Task-oriented gan for polsar image classification and clustering. *IEEE Trans. Neural Netw. Learn. Syst.* 30, 9 (2019), 2707–2719.
- [127] Donggeun Yoo, Namil Kim, Sunggyun Park, Anthony S. Paek, and In So Kweon. 2016. Pixel-level domain transfer. In *Proceedings of the European Conference on Computer Vision*. 517–532.
- [128] Shuchang Zhou, Taihong Xiao, Yi Yang, Dieqiao Feng, Qinyao He, and Weiran He. 2017. Genegan: Learning object transfiguration and attribute subspace from unpaired data. arXiv:1705.04932. Retrieved from <https://arxiv.org/abs/1705.04932>.
- [129] Nasim Souly, Concetto Spampinato, and Mubarak Shah. 2017. Semi-supervised semantic segmentation using the generative adversarial network. In *Proceedings of the IEEE International Conference on Computer Vision*. 5688–5696.
- [130] Shuyang Gu, Jianmin Bao, Hao Yang, Dong Chen, Fang Wen, and Lu Yuan. 2019. Mask-guided portrait editing with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3436–3445.
- [131] Yibing Song, Chao Ma, Xiaohe Wu, Lijun Gong, Linchao Bao, Wangmeng Zuo, Chunhua Shen, Rynson Lau, and Ming-Hsuan Yang. 2018. Vital: Visual tracking via adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8990–8999.
- [132] Yamin Han, Peng Zhang, Wei Huang, Yufei Zha, Garth Douglas Cooper, and Yanning Zhang. 2019. Robust visual tracking using unlabeled adversarial instance generation and regularized label smoothing. *Pattern Recogn.* (2019), 1–15.
- [133] Weidong Yin, Yanwei Fu, Leonid Sigal, and Xiangyang Xue. 2017. Semi-latent gan: Learning to generate and modify facial images from attributes. arXiv:1704.02166. Retrieved from <https://arxiv.org/abs/1704.02166>.
- [134] Chris Donahue, Zachary C. Lipton, Akshay Balsubramani, and Julian McAuley. 2018. Semantically decomposing the latent spaces of generative adversarial networks. In *Proceedings of the International Conference on Learning Representations*.
- [135] Aamanda Duarte, Francisco Roldan, Miquel Tubau, Janna Escur, Santiago Pascual, Amaia Salvador, Eva Mohedano, Kevin McGuinness, Jordi Torres, and Xavier Giro-i Nieto. 2019. Wav2pix: Speech-conditioned face generation using generative adversarial networks. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*.
- [136] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. 2019. Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1155–1164.
- [137] Zhixin Shu, Mihir Sahasrabudhe, Alp Guler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. 2018. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *Proceedings of the European Conference on Computer Vision*. 650–665.
- [138] Chaoyou Fu, Xiang Wu, Yibo Hu, Huaibo Huang, and Ran He. 2019. Dual variational generation for low-shot heterogeneous face recognition. In *Advances in Neural Information Processing Systems*.
- [139] Jie Cao, Yibo Hu, Bing Yu, Rai He, and Zhenan Sun. 2019. 3d aided duet gans for multi-view face image synthesis. *IEEE Trans. Inf. Forens. Secur.* 14, 8 (2019), 2028–2042.
- [140] Xuelin Qian, Yanwei Fu, Tao Xiang, Wenxuan Wang, Jie Qiu, Yang Wu, Y Gang, Jiang, and Xiangyang Xue. 2018. Pose-normalized image generation for person re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV'18)*. 650–667.
- [141] Jialun Liu, Wenhui Li, and Hongbin Pei. 2019. Identity preserving generative adversarial network for cross-domain person re-identification. *IEEE Access* (2019), 114021–114032.
- [142] Matteo Fabbri, Calderara Simone, and Cucchiara Rita. 2017. Generative adversarial models for people attribute recognition in surveillance. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*. 1–6.
- [143] Federico Fulgeri, Matteo Fabbri, Calderara Simone, and Cucchiara Rita. 2019. Can adversarial networks hallucinate occluded people with a plausible aspect? arXiv:1901.08097. Retrieved from <https://arxiv.org/abs/1901.08097>.
- [144] Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming-Ting Sun. 2017. Adversarial ranking for language generation. arXiv:1705.11001. Retrieved from <https://arxiv.org/abs/1705.11001>.
- [145] Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang. 2017. Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks. arXiv:1704.00849. Retrieved from <https://arxiv.org/abs/1704.00849>.
- [146] Olof Mogren. 2016. C-rnn-gan: Continuous recurrent neural networks with adversarial training. arXiv:1611.09904. Retrieved from <https://arxiv.org/abs/1611.09904>.
- [147] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* (1997).
- [148] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 2852–2858.

- [149] Gabriel Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. 2017. Objective reinforced generative adversarial networks (organs) for sequence generation models. arXiv:1705.10843. Retrieved from <https://arxiv.org/abs/1705.10843>.
- [150] Sudha Rao and Hai Daumé III. 2019. Answer-based adversarial training for generating clarification questions. arXiv:1904.02281. Retrieved from <https://arxiv.org/abs/1904.02281>.
- [151] Xiao Yang, Madian Khabsa, Miaosen Wang, Wei Wang, Ahmed Hassan, Awadallah, and Daniel Kifer, and Lee Giles. 2019. Adversarial training for community question answer selection based on multi-scale matching. arXiv:1804.08058. Retrieved from <https://arxiv.org/abs/1804.08058>.
- [152] Yong Luo, Huazizheng Zhang, Y. Yonggang Wen, and Xinwen Zhang. 2019. Resumegan: An optimized deep representation learning framework for talent job fit via adversarial learning. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*.
- [153] Cristina Garbacea, Samuel Carton, and Shiyuan Yan, and Qiaozhu Mei. 2019. Judge the judges: A large-scale evaluation study of neural language models for online review generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*. 2019.
- [154] Hojjat Aghakhani, Aravind Machiry, Shirin Nilizadeh, Christopher Kruegel, and Giovanni Vigna. 2018. Detecting deceptive reviews using generative adversarial networks. In *Proceedings of the IEEE Security and Privacy Workshops*, 89–95, 2018.
- [155] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 515–524.
- [156] Shuqi Lu, Zhicheng Dou, Jun Xu, Jian-Yun Nie, and Ji-Rong Wen. 2019. Psgan: A minimax game for personalized search with limited and noisy click data. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. 555–564.
- [157] Xin Wang, Wenhui Chen, Yuan-Fang Wang, and William Yang Wang. 2018. No metrics are perfect: Adversarial reward learning for visual storytelling. arXiv:1804.09160. Retrieved from <https://arxiv.org/abs/1804.09160>.
- [158] Santiago Pascual, Antonio Bonafonte, and Joan Serrà. 2017. Segan: Speech enhancement generative adversarial network. arXiv:1703.09452. Retrieved from <https://arxiv.org/abs/1703.09452>.
- [159] Chris Donahue, Bo Li, and Rohit Prabhavalkar. 2018. Exploring speech enhancement with generative adversarial networks for robust speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. 5024–5028.
- [160] Rakshith Shetty, Marcus Rohrbach, Lisa Anne Hendricks, Mario Fritz, and Bernt Schiele. 2017. Speaking the same language: Matching machine to human captions by adversarial training. In *Proceedings of the IEEE International Conference on Computer Vision*. 4135–4144.
- [161] Bei Liu, Jianlong Fu, Makoto P. Kato, and Masatoshi Yoshikawa. 2018. Beyond narrative description: Generating poetry from images by multi-adversarial training. In *Proceedings of the ACM Multimedia Conference on Multimedia Conference*. 783–791.
- [162] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Generating videos with scene dynamics. In *Advances in Neural Information Processing Systems*. 613–621.
- [163] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. 2018. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1526–1535.
- [164] Emily Denton and Vighnesh Birodkar. 2017. Unsupervised learning of disentangled representations from video. In *Advances in Neural Information Processing Systems*. 4414–4423.
- [165] Aidan Clark, Karen Simonyan, and Jeff Donahue. 2019. Adversarial video generation on complex datasets. arXiv: 1907.06571. Retrieved from <https://arxiv.org/abs/1907.06571>.
- [166] Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. 2017. The pose knows: Video forecasting by generating pose futures. arXiv:1705.00053. Retrieved from <https://arxiv.org/abs/1705.00053>.
- [167] Xiaodan Liang, Lisa Lee, Wei Dai, and Eric P. Xing. 2017. Dual motion gan for future-flow embedded video prediction. In *Proceedings of the IEEE International Conference on Computer Vision*. 1744–1752.
- [168] Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. 2018. Recyclegan: Unsupervised video retargeting. In *Proceedings of the European Conference on Computer Vision*. 119–135.
- [169] Zhou Zhao, Shuwen Xiao, Zehan Song, Chujie Lu, Jun Xiao, and Yuetong Zhuang. 2020. Open ended video question answering via multi-modal conditional adversarial networks. *IEEE Trans. Image Process.* (2020), 3859–3870.
- [170] Alexander. Y. Sun. 2018. Discovering state-parameter mappings in subsurface models using generative adversarial networks. *Geophys. Res. Lett.* 45, 20 (2018), 11–137.
- [171] Zhi Zhonga, Alexander. Y. Sun, and Xinming Wu. 2020. Inversion of time-lapse seismic reservoir monitoring data using cyclegan: A deep learning-based approach for estimating 3 dynamic reservoir property changes. *J. Geophys. Res.: Solid Earth* 125, 3 (2020).

- [172] Mario Rütters, Sangseung Lee, Soohwan Jeon, and Donghyun You. 2019. Prediction of a typhoon track using a generative adversarial network and satellite images. *Sci. Rep.* 9 (2019), 6057.
- [173] Kun Qian, Abdullaah Mohamed, and Christian Claudel. 2019. Physics informed data-driven model for flood prediction: Application of deep learning in the prediction of urban flood development. arXiv:1908.10312. Retrieved from <https://arxiv.org/abs/1908.10312>.
- [174] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*. 3856–3866.
- [175] Weiwei Hu and Ying Tan. 2017. Generating adversarial malware examples for black-box attacks based on gan. arXiv:1702.05983. Retrieved from <https://arxiv.org/abs/1702.05983>.
- [176] Muthuraman Chidambaram and Yanjun Qi. 2017. Style transfer generative adversarial networks: Learning to play chess differently. arXiv:1702.06762. Retrieved from <https://arxiv.org/abs/1702.06762>.
- [177] Han Shu, Yunhe Wang, Xu Jia, Kai Han, Hanting Chen, Chunjing Xu, Qi Tian, and Chang Xu. 2019. Co-evolutionary compression for unpaired image translation. In *Proceedings of the International Conference on Computer Vision*, 2019.
- [178] Shaohui Lin, Rongrong Ji, Chenqian Yan, Baochang Zhang, Liujuan Cao, Qixiang Ye, Feiyue Huang, and David Doermann. 2019. Towards optimal structured cnn pruning via generative adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2790–2799.
- [179] Yunchao Zhang, Y. Yanjie Fu, Pengyang Wang, Xiaolin Li, and Yu Zheng. 2019. Unifying inter-region autocorrelation and intra-region structures for spatial embedding via collective adversarial learning. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1700–1708.
- [180] Pengyang Wang, Yanjie Fu, Hui Xiong, and Xiaolin Li. 2019. Adversarial substructured representation learning for mobile user profiling. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 130–138.
- [181] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. 2018. Gan based synthetic medical image augmentation for increased cnn performance in liver lesion classification. *Neuro Comput.* 321 (2018), 321–331.
- [182] Junliang Yu, Hongzhi Yin*, Jundong Li, Min Gao, Zi Huang, and Lizhen Cui. 2019. Enhancing collaborative filtering with generative augmentation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 548–556.
- [183] Binbin Hu, Yuan Fang, and Chuan Shi. 2019. Adversarial learning on heterogeneous information networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 120–129.
- [184] Martin Abadi and David G. Andersen. 2016. Learning to protect communications with adversarial neural cryptography. arXiv:1610.06918. Retrieved from <https://arxiv.org/abs/1610.06918>.
- [185] Brett K. Beaulieu-Jones, Zhiwei Steven Wu, Chris Williams, Ran Lee, Sanjeev P. Bhavnani, James Brian Byrd, and Casey S. Greene. 2018. Privacy-preserving generative deep neural networks support clinical data sharing. BioRxiv: 159756.
- [186] Vahid Mirjalili, Sebastian Raschka, and Arun Ross. 2020. Privacynet: Semi adversarial networks for multi-attribute face privacy. arXiv:2001.00561. Retrieved from <https://arxiv.org/abs/2001.00561>.
- [187] Bo-Wei Tseng and Pei-Yuan Wu. 2020. Compressive privacy generative adversarial network. *IEEE Trans. Inf. Forens. Secur.* 15 (2020), 2499–2513.
- [188] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. 2018. Social gan socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2255–2264.
- [189] Aidan N. Gomez, Sicong Huang, Ivan Zhang, Bryan M. Li, Muhammad Osama, and Lukasz Kaiser. 2018. Unsupervised cipher cracking using discrete gans. arXiv:1801.04883. Retrieved from <https://arxiv.org/abs/1801.04883>.
- [190] Eder Santana and George Hotz. 2016. Learning a driving simulator. arXiv:1608.01230. Retrieved from <https://arxiv.org/abs/1608.01230>.
- [191] Michal Uricar, Pavel Krizek, David Hurich, Ibrahim Sobh, Senthil Yogamani, and Patrick Denny. 2020. Yes, we gan: Applying adversarial techniques for autonomous driving. arXiv:1902.03442. Retrieved from <https://arxiv.org/abs/1902.03442>.
- [192] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. arXiv:1705.08690. Retrieved from 2017 from <https://arxiv.org/abs/1705.08690>.
- [193] Alex Zhavoronkov. 2018. Artificial intelligence for drug discovery, biomarker development, and generation of novel chemistry. *Molec. Pharmaceut.* 15, 10 (2018).
- [194] Dmitry Efimov, Di Xu, Luyang Kong, Alexey Nefedov, and Archana Anandakrishnan. 2020. Using generative adversarial networks to synthesize artificial financial datasets. arXiv:2002.02271. Retrieved from <https://arxiv.org/abs/2002.02271>.

- [195] Kang Zhang, Guoqiang Zhonga, Junyu Donga, Shengke Wang, and Yong Wanga. 2018. Stock market prediction based on generative adversarial network. In *Proceedings of the International Conference on Identification, Information & Knowledge in Internet of Things*.
- [196] Magnus Wiese, Robert Knobloch, Ralf Korn, and Peter Kretschmer. 2019. Quant gans: Deep generation of financial time series. arXiv:190706673. Retrieved from <https://arxiv.org/abs/190706673>.
- [197] Xingyu Zhou, Zhisong Pan, Guyu Hu, Siqi Tang, and Cheng Zhao. 2018. Stock market prediction on high-frequency data using generative adversarial nets. *Math. Probl. Eng.* (2018).
- [198] Ashutosh Kumar, Arijit Biswas, and Subhajit Sanyal. 2018. “Ecommerce-gan: A generative adversarial network for e-commerce. arXiv:180103244. Retrieved from <https://arxiv.org/abs/180103244>.
- [199] Xingyu Zhou, Zhisong Pan, Guyu Hu, Siqi Tang, and Cheng Zhao. 2018. Stock market prediction on high-frequency data using generative adversarial nets. *Math. Probl. Eng.* (2018).
- [200] Nikolay Jetchev and Urs Bergmann. 2017. The conditional analogy gans: Swapping fashion articles on people images. arXiv:1709.04695. Retrieved from <https://arxiv.org/abs/1709.04695>.
- [201] You Xie, Erik Franz, Mengyu Chu, and Nils Thuerey. 2018. Tempo-gan: A temporally coherent, volumetric gan for super-resolution fluid flow. arXiv:1801.09710. Retrieved from <https://arxiv.org/abs/1801.09710>.
- [202] Maximilian Werhahn, You Xie, Mengyu Chu, and Nils Thuerey. 2019. A multi-pass gan for fluid flow super-resolution. arXiv:1906.01689. Retrieved from <https://arxiv.org/abs/1906.01689>.
- [203] Kyongsik Yun, Jessi Bustos, and Thomas Lu. 2018. Predicting rapid fire growth (flashover) using conditional generative adversarial networks. arXiv:1801.09804. Reference from <https://arxiv.org/abs/1801.09804>.
- [204] Zefeng Li, Men-Andrin Meier, Egill Hauksson, Zhongwen Zhan, and Jennifer Andrews. 2018. Machine learning seismic wave discrimination: Application to earthquake early warning. *Geophys. Res. Lett.* (2018).
- [205] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556. Retrieved from <https://arxiv.org/abs/1409.1556>.
- [206] Lillian J. Ratliff, Samuel A. Burden, and S. Shankar Sastry. 2013. Characterization and computation of local nash equilibria in continuous games. In *Proceedings of the Annual Allerton Conference on Communication, Control and Computing*. 917–924.
- [207] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167. Retrieved from <https://arxiv.org/abs/1502.03167>.
- [208] Ian Goodfellow. 2016. Nips 2016 tutorial: Generative adversarial networks. arXiv:1701.00160. Retrieved from <https://arxiv.org/abs/1701.00160>.
- [209] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. 2017. Generalization and equilibrium in generative adversarial nets (gans). In *Proceedings of the International Conference on Machine Learning*. 224–232.
- [210] Ali Borji. 2019. Pros and cons of gan evaluation measures. *Comput. Vis. Image Understand.* 179 (2019), 41–65.
- [211] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *Neural Information Processing Systems* 2234–2242, 2016.
- [212] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. 2016. Unrolled generative adversarial networks. arXiv:1611.02163. Retrieved from <https://arxiv.org/abs/1611.02163>.
- [213] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gan trained by a two time-scale update rule converge to local nash equilibrium. In *Advances in Neural Information Processing Systems*. 6626–6637.
- [214] Anders Boesen Lindbo Larse, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. 2015. Autoencoding beyond pixels using a learned similarity metric. arXiv:1512.09300. Retrieved from <https://arxiv.org/abs/1512.09300>.
- [215] Jin-Young Kim, Seok-Jun Bu, and Sung-Bae Cho. 2018. Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders. *Inf. Sci.* 460 (2018) 83–102.
- [216] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. 2018. Self-attention generative adversarial networks. arXiv:1805.08318. Retrieved from <https://arxiv.org/abs/1805.08318>.
- [217] Ashish Vaswani, Noam Shazeer, Niki Parmar, Llion Jones, Aidan N. Gomez, and Lukasz Kaiser. 2017. Attention is all you need. arXiv:1706.03762. Retrieved from <https://arxiv.org/abs/1706.03762>.
- [218] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. 2018. Image transformer. arXiv:1802.05751. Retrieved from <https://arxiv.org/abs/1802.05751>.
- [219] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-local neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 779–788.
- [220] Jolicoeur-Martineau. 2019. The relativistic discriminator: A key element missing from standard gan. In *Proceedings of the International Conference on Learning Representation*.
- [221] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2818–2826.

- [222] Tom White. 2016. Sampling generative networks. arXiv:1609.04468. Retrieved from <https://arxiv.org/abs/1609.04468>.
- [223] Tijmen Tieleman and Geoffrey E. Hinton. 2012. Divide the gradient by a running average of its recent magnitude. *Neur. Netw. Mach. Learn.* 4, (2012).
- [224] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. 2018. A mean-field view of the landscape of two-layer neural networks. arXiv:1804.06561. Retrieved from <https://arxiv.org/abs/1804.06561>.
- [225] John Duchi, Elad Hazan, and Yoram Singer. 2012. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* (2012), 2121–2159.
- [226] Diederik P. Kingma, and Jimmy Lei Ba. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980. Retrieved from <https://arxiv.org/abs/1412.6980>.
- [227] Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. 2017. Amortised map inference for image super-resolution. In *Proceedings of the International Conference on Learning Representations*.
- [228] Martin Arjovsky and Leon Bottou. 2017. Towards principled methods for training generative adversarial networks. In *Proceedings of the International Conference on Learning Representations*. 214–223.
- [229] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. 2017. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 2794–2802.
- [230] Duhyeon Bang and Hyunjung Shim. 2018. Improved training of generative adversarial networks using representative features. In *Proceedings of the International Conference on Machine Learning*.
- [231] R. Devon Hjelm, Athul Paul Jacob, Tong Che, Adam Trischler, Kyunghyun Cho, and Yoshua Bengio. 2018. Boundary-seeking generative adversarial networks. In *Proceedings of the International Conference on Learning Representations*.
- [232] Tong Che, Yanran Li, Ruixiang Zhang, R. Devon Hjelm, Wenjie Li, Yangqiu Song, and Yoshua Bengio. 2017. Maximum-likelihood augmented discrete generative adversarial networks. arXiv:1702.07983. Retrieved from <https://arxiv.org/abs/1702.07983>.
- [233] Jake Zhao, Yoon Kim, Kelly Zhang, Alexander M. Rush, and Yann LeCun. 2017. Adversarial regularized autoencoders for generating discrete structures. arXiv:1706.04223. Retrieved from <https://arxiv.org/abs/1706.04223>.
- [234] Henning Petzka, Asja Fischer, and Denis Lukovnikov. 2017. On the regularization of wasserstein gans. arXiv:1709.08894. Retrieved from <https://arxiv.org/abs/1709.08894>.
- [235] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. In *Proceedings of the International Conference on Learning Representations*.
- [236] Mikołaj Binkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. 2018. Demystifying mmd-gans. In *Proceeding of the International Conference on Learning Representation*.
- [237] Zhou Wang, Eero P. Simoncelli, and Alan C. Bovik. 2003. Multiscale structural similarity for image quality assessment. In *Proceedings of the Asilomar Conference on Signals, Systems & Computers*. 1398–1402.
- [238] Erich L. Lehmann and Joseph P. Romano. 2006. *Testing Statistical Hypotheses*. Springer, New York, NY.
- [239] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *J. Mach. Learn. Represent.* 13, 1 (2012), 723–773.
- [240] Ilya Tolstikhin, Sylvain Gelly, Olivier Bousquet, Carl-Johann Simon-Gabriel, and Bernhard Schölkopf. 2017. AdaGAN: Boosting generative models. In *Advances in Neural Information Processing Systems*, 5424–5433, 2017.
- [241] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. 2016. Mode regularized generative adversarial networks. arXiv:1612.02136. Retrieved from <https://arxiv.org/abs/1612.02136>.
- [242] Arnab Ghosh, Viveka Kulharia, Vinay Namboodiri, Philip H. S. Torr, and Puneet K. Dokania. 2017. Multi-agent diverse generative adversarial networks. arXiv:1704.02906. Retrieved from <https://arxiv.org/abs/1704.02906>.
- [243] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. arXiv:1607.06450. Retrieved from <https://arxiv.org/abs/1607.06450>.
- [244] Tim Salimans and Diederik P. Kingma. 2016. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. arXiv:1602.07868. Retrieved from <https://arxiv.org/abs/1602.07868>.
- [245] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2016. Instance normalization: The missing ingredient for fast stylization. arXiv:1607.08022. Retrieved from <https://arxiv.org/abs/1607.08022>.
- [246] Yuxin Wu and Kaiming He. 2018. Group normalization. arXiv:1803.08494. Retrieved from <https://arxiv.org/abs/1803.08494>.
- [247] Hyeonseob Nam and Hyo-Eun Kim. 2018. Batch instance normalization for adaptively style invariant neural networks. arXiv:1805.07925. Retrieved from <https://arxiv.org/abs/1805.07925>.
- [248] Ping Luo, Jiamin Ren, Zhanglin Peng, Ruimao Zhang, and Jingyu Li. 2019. Differentiable learning to normalize via switchable normalization. In *Proceedings of the International Conference on Learning Representations (ICLR'19)*.
- [249] Alex Krizhevsky and Geoffrey Hinton. 2009. Learning multiple layers of features from tiny images. Citeseer, Tech. Rep.

Received March 2020; revised March 2021; accepted April 2021