Artificial Intelligence in the Life Sciences 2 (2022) 100045

Contents lists available at ScienceDirect

Artificial Intelligence in the Life Sciences

journal homepage: www.elsevier.com/locate/ailsci

Review

Recent advances and application of generative adversarial networks in drug discovery, development, and targeting

Satvik Tripathi[a,*], Alisha Isabelle Augustin[b], Adam Dunlop[c], Rithvik Sukumaran[a], Suhani Dheer[c], Alex Zavalny[a], Owen Haslam[c], Thomas Austin[d], Jacob Donchez[e], Pushpendra Kumar Tripathi[f], Edward Kim[a]

[a] College of Computing and Informatics Drexel University Philadelphia, PA 19104 USA
[b] College of Engineering Drexel University Philadelphia, USA
[c] College of Arts and Sciences Drexel University Philadelphia, PA 19104 USA
[d] College of Engineering Drexel University Philadelphia, PA 19104 USA
[e] College of Biomedcial Engineering Drexel University Philadelphia, PA 19104 USA
[f] Institute of Pharmaceutical Sciences University of Lucknow Lucknow, India

ARTICLE INFO

Keywords:
Generative adversarial networks
Machine learning
Artificial intelligence
Pharmacology
Drug discovery
Drug targeting

ABSTRACT

A rising amount of research demonstrates that artificial intelligence and machine learning approaches can provide an essential basis for the drug design and discovery process. Deep learning algorithms are being developed in response to recent advances in computer technology as part of the creation of therapeutically relevant medications for the treatment of a variety of ailments. In this review, we focus on the most recent advances in the areas of drug design and discovery research employing generative deep learning methodologies such as generative adversarial network (GAN) frameworks. To begin, we examine drug design and discovery studies that use several GAN methodologies to evaluate one key application, such as molecular *de novo* design in drug design and discovery. Furthermore, we discuss many GAN models for dimension reduction of single-cell data at the preclinical stage of the drug development pipeline. We also show various experiments in *de novo* peptide and protein creation utilizing GAN frameworks. Furthermore, we discuss the limits of past drug design and discovery research employing GAN models. Finally, we give a discussion on future research prospects and obstacles.

1. Introduction

In recent years, researchers have made remarkable strides in the interconnected domains of artificial intelligence, machine learning, as well as drug design and discovery [1–4]. The purpose of artificial intelligence and machine learning approaches in the field of drug design and discovery is to develop data-driven algorithms that, in general, can help facilitate various stages of the drug development pipeline, such as drug target prediction, drug screening, and discovery, pre-clinical trials, and clinical trials [2,5,6]. The most recent developments in artificial intelligence and machine learning technologies, in particular, deep learning algorithms [7,8], have revealed their potentially exciting possibilities in relation to drug design and discovery [1–10,12–14]. For example, in the preclinical stage of the drug development pipeline, deep learning approaches such as deep variational autoencoder [15] have been used to conduct the dimension reduction task of single-cell data for cell-specific biomarker discovery using single-cell RNA sequencing (scRNA-seq) techniques [16,17]. In this way, single-cell data for cell-specific biomarker discovery have been simplified. In addition, the synthesis of new chemical structures through the use of deep variational autoencoder during the drug screening and discovery stage is yet another fascinating example of the application of deep learning methodologies [18]. Because of this, it has been hypothesized that deep learning techniques will play a crucial part in the future of drug design and discovery. This is due to the fact that the relevant applications of these approaches include a wide range of facets related to drug design and discovery [3,19].

Deep learning techniques, in their most basic form, combine various forms of sophisticated artificial intelligence with machine learning models. These models employ a number of different levels of abstraction in order to construct hierarchical representations of the data [20–22]. For instance, artificial neural networks can be used to construct the hierarchical representation [22,23]. To put it another way, deep learning techniques are computer programs that resolve the best predictions by employing artificial neural networks that include several layers, as op-

* Corresponding author.
E-mail address: st3263@drexel.edu (S. Tripathi).

https://doi.org/10.1016/j.ailsci.2022.100045
Received 31 July 2022; Received in revised form 16 October 2022; Accepted 25 October 2022
Available online 31 October 2022

posed to employing artificial neural networks that only employ a single individual layer [22]. Deep learning algorithms have achieved a broad variety of applications in drug design and discovery [9,21]. These approaches are based on the most advanced computer technologies now available, such as general-purpose computing performed on graphics processing units. There is an enormous need to employ software tools in deep learning frameworks for various drug development tasks [2] in order to address the demanding challenges we face today in the field of drug design and discovery. These challenges are presented in the form of complex problems that must be solved. To be more specific, deep learning frameworks are utilized in order to serve as tools in order to fulfill the applications of drug design and discovery, such as molecular *de novo* design, dimension reduction of single-cell data in pre-clinical development, compound property and activity prediction, reaction analysis, synthesis prediction, and biological image analysis [1]. These applications include molecular *de novo* design, dimension reduction of single-cell data in pre-clinical development, reaction analysis, synthesis prediction, and biological image analysis.

An up-and-coming method known as the generative adversarial network (GAN) architecture [24] has been garnering a growing amount of interest in both the fields of artificial intelligence and machine learning research as a result of recent developments in deep learning frameworks. To begin, the GAN architecture has a significant potential to be utilized in a wide variety of applications, including the creation and discovery of new drugs, the analysis of photos and videos, the translation of language, and other areas [25–28]. In addition, the implementation of the GAN architecture has been making a contribution to the study on drug design and discovery. Recent years have seen a broad range of important research investigations for the design and discovery of new drugs, such as molecular *de novo* design, which took into account the GAN architecture [2,9]. For instance, GAN-based frameworks such as the deep adversarial autoencoder structure have been used to design and find new drugs for anticancer therapy by utilizing chemical and biological datasets [29,30]. This has been accomplished by combining the two types of data. In addition, the deep adversarial variational autoencoder structure has been shown to successfully complete the task of dimensionality reduction for single-cell RNA sequencing data in the preclinical stage of the drug development pipeline [28]. This is a remarkably intriguing example for a number of reasons, not the least of which is the fact that it fulfills the task. In the following sections, we will elaborate on the intricacies of several GAN-based frameworks that are used in drug design and discovery. Some examples of these frameworks include the deep adversarial autoencoder and the deep adversarial variational autoencoder structures.

In this article, we review a variety of research studies that are focused on three major categories in terms of drug design and discovery [31–35]. These categories are molecular *de novo* design, dimension reduction of single-cell data in preclinical development, and *de novo* peptide and protein design. All of these research studies are presented within the context of GAN-based frameworks [36,37]. Because, to the best of our knowledge, there may not be many studies in drug design and discovery that use the GAN-based frameworks for other applications at the time of the submission of this paper, we primarily focus on these three applications employing a broad variety of the GAN-based frameworks. In light of this, the biological and/or therapeutic implications stemming from these three key sectors might potentially serve as a platform for future study in the design and discovery of drugs utilizing GAN-based frameworks [38–41]. In addition to this, we describe the restrictions that were placed on these research investigations and provide a summary of a debate regarding the future difficulties and directions. This review does not support the full set of related research studies that were reported in the literature [42–44]; however, it does describe a synthesis of those studies that have the potential to significantly influence public and population health-oriented applications in drug design and discovery using GAN-based frameworks in the relatively near to intermediate future.

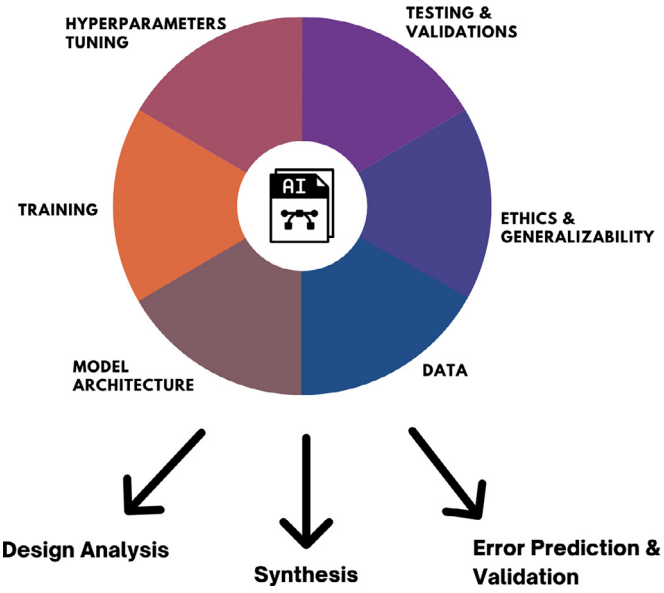

Fig. 1. Key Components of Artificial Intelligence and Useful Outcomes in Drug and Molecular Modeling.

2. Artificial intelligence

Artificial intelligence (AI) is described as a scientific and engineering subject concerned with the computational understanding of what is usually referred to as "intelligent behavior," as well as the production of artifacts that display such behavior. Through his syllogisms, Aristotle endeavored to systematize 'proper thinking' (logic) (three-part deductive reasoning). Much of the work in the modern age was inspired by this, and early research on the workings of the mind contributed to the development of modern logical reasoning. Artificial intelligence systems are programs that allow computers to work in ways that make people appear intelligent. Alan Turing (1950), a British mathematician, was a pioneer of contemporary computer science and artificial intelligence. He characterized intelligent behavior in a computer as the ability to do cognitive activities at the level of a person. This became known as the 'Turing test' later on. Researchers have been investigating the possible uses of intelligent approaches in every aspect of medicine since the mid-twentieth century. Gunn initially researched the application of AI technology in the field of surgery in 1976, when he investigated the potential of detecting severe abdominal discomfort with computer analysis. Medical AI has experienced a boom in interest during the previous two decades (Fig. 1).

Machine Learning is defined as an assortment of complex algorithms designed to emulate that of the human mind through learning from the surrounding environment and/or experiences. When working on machine learning, there are three focuses of the model: the class of the task, the factor by which the performance must be improved by, and the source of the overall experience [45]. Using these three factors, machine learning models have the optimization goal of minimizing their learning error through the use of a variety of algorithms such as gradient descent and backpropagation. Gradient descent looks at the error curve that the model generates compared to what it should be outputting and finds the right parameters for the model that minimizes the error. These algorithms have been applied in a very broad range of environments consisting of, but not limited to: computer vision, entertainment, spacecraft engineering, finance, and computational biology [45].

There are three main types of machine learning to discuss: supervised, unsupervised, and reinforcement learning. Supervised learning is trained on labeled data, and is used to make predictions on new data given. The predictions can come in the form of classification with out-

putting a discrete number of classes or regression outputting a continuous value or set of values [46]. For example, a machine learning model such as linear regression classifying a given image as containing a cat or a dog would first need to be separately shown images of a cat and a dog to learn the features within the images to then compare to the features of a new image. For regression, a model could be trained on the previous sales of a company to then create a numerical prediction to forecast future sales and growth of the next year.

Unsupervised learning uses unlabeled data clusters and creates associations among the data points in order to identify any similarities and sequences of repeating patterns among the data [47]. For example, if a company wants to segment its customer base into a range of clients from low to high priority, then it would use an unsupervised learning model such as a gaussian mixture model to market its product towards a certain customer segment in a given season. Another important use of unsupervised learning is to prepare an unlabeled dataset for a supervised learning task which is a hybrid task called self-supervised learning.

Both supervised and unsupervised learning involves the optimization of each model's parameters, their weights, and other hyperparameters, in order to minimize the error the model produces. However, reinforcement learning involves training a model that interacts with its environment in order to maximize a reward. The reinforcement agent is rewarded for desirable behaviors, such as an autonomous agent being able to walk properly, and is punished for undesirable behaviors such as the agent falling [48].

Deep learning is defined as a multi-layered learning module, referred to as a neural network. A neural network is composed of non-linear modules, transferring and converting the information between its layers. These non-linear modules are called activations which, similarly to a biological neuron's activation potential, determines how strongly information passes through the network. The activations in a neural network are determined by activation functions, which are subsequently determined as a parameter in the system [49]. Backpropagation is the popular process of training neural networks by defining a function, and using the hill-climbing technique to optimize performance on a set task. This optimization calculates the mathematical gradient of a loss function with respect to the neural network's other weights. Backpropagation has become very popular due to its simplistic implementation, and the ability to train non-linear networks of arbitrary connectivity [50].

One of the first known works of artificial intelligence takes place in the late eighteenth century, with the creation of æThe Turkg, an automated chess-playing machine that would be able to compete against its opponents [51]. It would not be for over 150 years, until 1942, that one can see another example of AI, æThe Bombeg. Alan Turning created æThe Bombeg, an electromechanical device, to aid the British in breaking German codes towards the end of World War II. From this point, the beginning of computation, and thus, AI took off [52]. In 1950, Alan Turing published the æTuring Testg, which is detailed in the definition of AI, as seen above [53]. Six years later, the æbeginning of AI springg occurred at the Dartmouth Summer Research Project on Artificial Intelligence (DSRPAI). This project lasted over the course of eight weeks, with the final goal being to gather researchers and create an area that focuses on building machines that simulate the human experience.

Upon the end of the project, it was considered a success within the world of artificial intelligence [54]. Soon after this, the machine learning program æperceptrong was created in 1958, in the Cornell Aeronautical Laboratory. This program was originally designed as an image recognition service, but is now used as an algorithm for binary classifiers; binary classifiers determine whether an input belongs to a specific class [55]. Five years later, in 1963, Feigenbaum and Feldman published æComputers and Thoughtg, which describes the internal workings of AI programs at a fairly basic level. This publication was influential in spreading the word about artificial intelligence. Following this, it took another 35 years for any other significant impacts to be made within the field [56]. In 1998, researchers were able to develop a revolutionary new way of using trainable filters to learn patterns from images. This program, coined the Convolutional Neural Network (CNN), allows AI to work much easier with image recognition, ending any possible interference from pixel resolution [57].

This concept of a CNN was further developed, as in 2012 the winner of the ImageNet Large Scale Visual Recognition Challenge was a program called AlexNet. This program used a deep convolutional neural network to win, achieving a model error of only 15.3 percent. Another revolutionary concept utilized within this program was that AlexNet utilized GPUs during training to show modern-day AI models can become more powerful with the inclusion of GPUs [58]. Then, two years later in 2014, Ian Goodfellow created the GAN, a type of machine learning in which the goal is: given a training set, the program can create new data with the overall same statistics as the old learning set. GANs have since been used in fashion, art, and advertising, science, video games, and is the underlying process that is used to generate deepfakes, which are often applied within medicine [59]. The following year, Google created the AlphaGo system, which is an artificial neural network that plays the board game æGog. This program utilizes a Monte Carlo tree search algorithm to find its moves based on previously acquired knowledge. The newest version of AlphaGo is currently ranked as the best player in the world at æGog [60]. Google then continued its work in neural networks into 2017 to improve the accuracy of their language translations. This approach uses a constant number of model parameters, leading to a much simpler process than any previous iteration [61]. Google's efforts with AI do not end there, as in 2021, Google DeepMind created the program AlphaFold, which was a key component in determining the shape of a 3D protein. On top of this, it outperformed 100 other research teams in a protein structure prediction challenge, showing how influential this new age of AI can be [62] (Fig. 2).

The challenge of modern medicine is obtaining, evaluating, and using a significant quantity of knowledge required to treat complicated clinical issues. The advancement of medical artificial intelligence has been linked to the creation of AI algorithms designed to assist doctors in the formulation of a diagnosis, the formulation of treatment decisions, and the prediction of results. They are intended to aid healthcare staff with activities that require the processing of data and expertise in their daily jobs. Deep neural networks (DNNs), Unets, fuzzy expert systems, evolutionary computation, and hybrid intelligent systems are examples of such systems.

AI has the potential to fundamentally alter medicine in the next few years. The field of medical AI has advanced significantly in just a few years since the first landmark demonstrations of medical AI algorithms that were able to diagnose a disease from medical pictures at the level of specialists. Today, while the medical AI community navigates the difficult ethical, technological, and human-centered obstacles necessary for safe and successful translation, the deployment of medical AI systems in everyday clinical care represents a significant but largely unrealized potential.

3. Recent progress in the application of AI in medicine

Despite the fact that AI systems have frequently been proven to be successful in a wide range of retrospective medical investigations, only a small number of AI tools have been applied to medical practices. Critics argue that AI systems may be less beneficial in practice than retrospective data would suggest; systems may be too sluggish or difficult to be effective in real-world medical settings, or unexpected issues may occur from the way humans and AIs interact. Furthermore, retrospective in silico datasets is subjected to intensive filtering and cleaning, making them less representative of real-world medical practice.

RCTs and prospective studies help bridge the gap between theory and practice by more thoroughly establishing that AI models can have a quantifiable, beneficial impact when used in real-world healthcare settings. RCTs have recently been conducted to assess the use of AI systems in healthcare. Aside from accuracy, a range of additional criteria has been employed to assess AI's value, offering a comprehensive picture

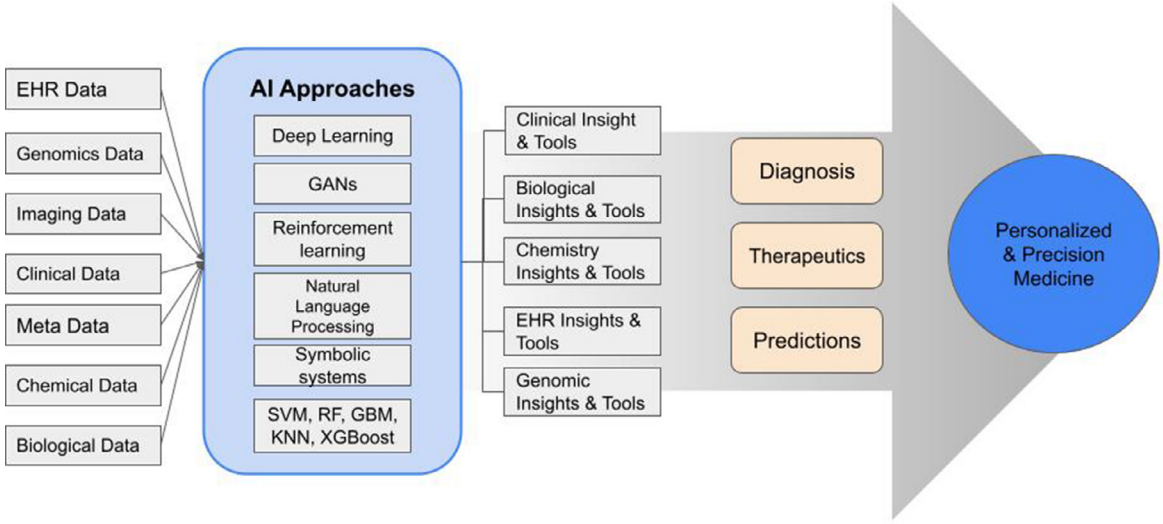

Fig. 2. Applications of Artificial Intelligence in Healthcare [63].

of its influence on medical systems (9–13). An RCT evaluating an AI system for managing insulin doses, for example, tracked the number of time patients spent within the target glucose range; a study evaluating a monitoring system for intraoperative hypotension tracked the average duration of hypotension episodes; a system that flagged cases of intracranial hemorrhage for human review was judged on its turnaround time.

Recent guidelines, such as AI-specific extensions to the SPIRIT and CONSORT guidelines, as well as upcoming guidelines like STARD-AI, may help standardize medical AI reporting, including clinical trial protocols and results, making it easier for the community to share findings and rigorously investigate the utility of medical AI17,18. Some AI systems have progressed from testing to implementation in recent years, gaining administrative backing and resolving regulatory difficulties. The Centers for Medicare and Medicaid Services, which oversees public insurance payment fees, has made AI more accessible in clinical settings by enabling funding for the use of two particular AI systems for medical image diagnosis.

Furthermore, a 2020 research discovered that the US Food and Drug Administration (FDA) is rapidly approving AI goods, notably machine learning. These advancements primarily take the form of FDA clearances, which require devices to achieve a lower regulatory threshold than full-fledged approvals, but they do pave the way for AI/ML systems to be utilized in genuine clinical situations. It should be noted that the datasets utilized for these regulatory approvals are frequently made up of retroactive, single-institution data that is mainly unpublished and deemed proprietary. Stronger requirements for reporting openness and validation, as well as evidence of influence on clinical outcomes, will be essential to develop confidence in medical AI systems.

Significant advances in pathology, mostly through the use of whole-slide imaging, have been made in identifying malignancies and offering novel disease insights. Models have successfully identified regions of interest within slides, possibly speeding up diagnostic operations. Aside from this practical impact, deep neural networks have been trained to distinguish the main tumor origin and find structural variations or driver mutations, delivering benefits that outperform expert pathologist evaluations. Furthermore, when compared to conventional grading and histological Through subtyping, AI has been demonstrated to produce more accurate survival forecasts for a wide variety of cancer types. Such research has shown how AI may improve the efficiency, accuracy, and utility of pathology diagnosis.

Deep learning has also advanced in gastroenterology, particularly in terms of enhancing colonoscopy, a critical test for detecting colorectal cancer. Deep learning has been used to predict if colonic lesions are cancerous, with results equivalent to those of competent endoscopists. Furthermore, because polyps and other potential indicators of illness are usually overlooked during the exam, artificial intelligence (AI) technologies have been created to aid endoscopists. Such technologies have been demonstrated to increase endoscopists' capacity to identify anomalies, potentially boosting sensitivity and making colonoscopy a more reliable diagnostic tool.

Deep learning models have been widely used in ophthalmology, with significant progress toward deployment. In addition to evaluating model effectiveness, research has looked into the human impact of such models on healthcare systems.

One study used human observation and interviews to assess how an AI system for eye disease screening influences patient experience and medical workflows. Other studies have investigated the financial impact of AI in the ophthalmology field, discovering that semi-automated or completely automated AI screening may provide cost savings in some situations, such as the identification of diabetic retinopathy.

4. Rise of GANs

4.1. Definition

Generative modeling is an unsupervised learning technique that captures the probability distribution $p(X)$ of some sample data X by matching the probability distribution of our model to the probability distribution of the sample data. The two main ways this is done is by learning an approximation of the function $p(X)$ that models the probability distribution through density estimation. The second technique involves similarly learning a probability distribution to generate new data samples $\hat{X}$ that is compared to the actual data X. Either way, modeling a probability distribution becomes increasingly complex as different data types such as images and text typically have high dimensional distributions that become difficult to model. That said, generative models are capable of uncovering the underlying features of a dataset in a completely unsupervised way.

Generative models can also be applied to classification such as its architecture in the Naive Bayes Classifier, the generative model learns the joint probability distribution $p(X, Y)$ of the input X and the label Y by using Baye's Theorem to calculate $p(Y|X)$ [64].

Where $p(Y|X)$ is found through:

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)} \tag{1}$$

The goal of a generative classifier such as the Naive Bayes Classifier is to then use the calculated $p(Y|X)$ to predict the most likely label Y for a given input X.

Discriminative modeling is a supervised learning technique that also uses conditional probability to find $p(Y|X)$, but instead does so through looking for a direct mapping from X to the data labels Y [64]. The key difference between discriminative and generating modeling is that instead of modeling the conditional probability of the data and generating new samples, the goal of discriminative modeling is to separate one class from another by creating decision boundaries. Logistic Regression is an example of a discriminative model that uses a decision boundary in the form of a line to separate the data into two parts for binary classification and outputs the probability of a data point belonging to one of two classes.

In a study comparing discriminative and generative models used for classification tasks on various datasets from the University of California Irvine Machine Learning Repository, A. Ng et al found that discriminative models such as logistic regression tend to have lower errors after training is complete but generative models such as Naive Bayes tend to converge much faster and result in shorter training times [64]. Would it be possible to avoid having to deal with the tradeoff of both types of models by combining the efforts of both of these distinct styles? The need for a hybrid model arises from attempting to model data with complex distributions. Modeling a distribution directly is often impossible since approximating the distributions of advanced data types such as images, audio and text involves too many probabilistic computations to scale with larger amounts of data [59].

In 2014, Deep Learning researcher Ian Goodfellow et al created a revolutionary new unsupervised learning model, the Generative Adversarial Network (GAN), that involves a generative model competing against a discriminative model as adversaries to produce new data samples that can't be distinguished from the training data [59]. When random noise, a simple distribution to model, is passed through a multilayer perceptron, the generator learns the transformation from the random noise distribution to the target data distribution. The discriminator is trained to distinguish the real samples from those generated by the generator. The optimization goal of the generator is to maximize the probability of the discriminator detecting its generated samples as true, which is when the discriminator makes a mistake, and the goal of the discriminator is to correctly classify the generated samples between the true data and the fake data generated by the generator. Training is stopped when the generated samples from the generator can't be distinguished from the real samples by the discriminator.

One of the biggest inspirations for GANs came from deep Boltzman machines that likewise utilized two separate processes running at the same time while training which included a positive and negative phase [59]. The positive phase loaded in data and made it more likely, and the negative phase drew samples and made them less likely. Through iterative training, the distribution of the samples becomes closer to the distribution the model represents. The parameters of the model are updated at the same time as new samples are being generated. However, it was found that deep Boltzman machines lacked the capability to scale past the greyscale MNIST digits dataset, and for example, were unable to generate color image samples in a timely manner. This is caused by the negative phase being unable to keep up with the speed of the positive phase generating samples which prevents deep Boltzman machines from converging to a useful state.

Since multilayer perceptrons are used for the discriminative and generative models, both models can be trained using backpropagation and are typically trained with dropout regularization [59]. For the discriminator, since it's being trained as a classification task, the binary cross entropy loss function is used with the true label y and the predicted label $\hat{y}$:

$$L(\hat{y}, y) = -ylog(\hat{y}) - (1-y)log(1-\hat{y}) \tag{2}$$

Over N instances, the equation becomes:

$$L(\hat{y}, y) = -\frac{1}{N}\sum_{n=1}^{N} y_n log(\hat{y_n}) + (1-y_n)log(1-\hat{y}_n) \tag{3}$$

Where ∇L is calculated and the hyperparameters of the generator and discriminator are adjusted accordingly to minimize the loss function. In these expressions, $\hat{y}$ is not a constant, and is the output of an activation function $\hat{y} = a(z)$ where $a(z)$ would typically be the ReLU or sigmoid activation functions $ReLU(z) = \max(0, z)$ and $\sigma(z) = \frac{1}{1+e^{-z}}$ respectively. Within each neuron and each weight in the discriminator neural network, we define a function $z(w, x, b) = wx + b$ that outputs the activation of that particular neuron. In practice, the weights and biases are vectorized. In order for gradient descent to be performed, the gradient of the loss function with respect to the weights are computed via the chain rule:

$$\frac{\partial L(\hat{y}, y)}{\partial w} = \frac{\partial L(\hat{y}, y)}{\partial a(z)}\frac{\partial a(z)}{\partial z}\frac{\partial z}{\partial w} \tag{4}$$

Where the gradient of the loss function with respect to the activation function is calculated as follows:

$$\frac{\partial L(\hat{y} = a(z), y)}{\partial \hat{y}} = -\frac{y}{a(z)} + \frac{1-y}{1-a(z)} \tag{5}$$

and the gradient of the activation function with respect to the weights is

$$\frac{\partial z}{\partial w} = \frac{\partial}{\partial x}(wx + b) = x \tag{6}$$

which results in for whatever activation function $a(z)$ is chosen

$$\frac{\partial L(\hat{y}, y)}{\partial w} = \left(-\frac{y}{a(z)} + \frac{1-y}{1-a(z)}\right)\frac{\partial a(z)}{\partial z}x \tag{7}$$

and is then finally used in the gradient descent algorithm to update the weights of the discriminator using η as the learning rate.

$$w = w - \eta\frac{\partial L(\hat{y}, y)}{\partial w} \tag{8}$$

As for training the generator, it must be alternated as the discriminator trains. The generator's loss comes from the penalization of the discriminator correctly classifying its generated sample as fake. As backpropagation flows from the output of the discriminator through the discriminator itself, it then additionally will flow into the generator so that the generator's weights will be updated without changing the discriminator's weights. Then in the next epoch, backpropagation flows through the discriminator and the discriminator's weights are updated while keeping the generator's weights constant. This process repeats until the discriminator's performance becomes 50% accurate which indicates that the generator is able to generate realistic samples of the data that the discriminator can't tell between the fake, generated samples and the real ones.

It's important to note that the discriminator portion of the GAN model has been applied to more broad classification problems with multiple classes as opposed to only determining if it's real or fake. Using semi-supervised learning results in an ability to train with far fewer training labels than previous and faster convergence.

GANs have been used in many different applications, but most notably in the generation of new, artificial faces. This can then be used to create videos of famous figures that are completely artificially generated but are realistically looking to fool the human eye. In the wrong hands, GANs can be used to create and promote fake news with artificially generated content about real people, otherwise known as deepfakes, to potentially be used to destroy the reputations of said people. Thus, it's vital that this powerful deep learning technology be used responsibly for the good of humanity.

4.1.1. Architecture

Generative Adversarial Networks (GANs) are comprised of two neural networks: Generator and Discriminator. The generator generates

plausible data, and the discriminator distinguishes between what part of the data is real or fake. The generator uses the feedback from the discriminator to update its weights and biases in order to generate data that can better fool the discriminator by looking more real. The discriminator will then begin to perform worse on the data and will have to update itself to perform better. After this, the generator will be able to use the feedback from the new discriminator to better improve its own data generation. This cycle continues until convergence. For image generation applications, the discriminator is typically a Convolutions Neural Network (CNN) and is able to adapt to the underlying distribution of data. It will perform a binary classification to distinguish between the real and the generated data.

After Goodfellow et al published the Generative Adversarial Nets paper, the Deep Learning research community became amazed at how efficiently higher-quality samples were generated. Soon after, Mirza et al modified the GAN architecture by feeding the input data into both the discriminator and generator, and created the Conditional GAN (CGAN) [65]. The CGAN allows you to dictate what type of data should be generated with a condition based on class labels or any modality. In application, this can be used to identify which parts of the road are important for an autonomous driving vehicle to avoid that were not part of the original training environment, or what sections of a brain scan are signs of a disease external to the current experiment (Fig. 3).

A year later, Denton et al combined the CGAN with a Laplacian pyramid to break up the generation of images through multiple CNNs into successive refinements [66]. The Laplacian GAN (LapGAN) was the first GAN architecture to be able to generate high-resolution images by first generating images at a low resolution and then scaling to successive higher resolutions. This was done by taking the difference between the generated images and their blurred counterparts to produce a learnable filter.

A few months later, Radford et al were trying to integrate the CNN architecture, previously used in supervised learning, into the unsupervised learning goal of GANs in order to better suit computer vision tasks. They solved their issues of scaling the CNN architectures with GANs by replacing all of the pooling layers with strided and fractional strided convolutional layers [67]. Additionally, the fully connected layers that typically connected the final convolutional layers to the output neurons are removed. The generator takes in input samples and generates convolutional filters from which the discriminator deconvolves the filters into an image and classifies the generated image. This is called a Deep Convolutional GAN (DCGAN), and what makes this architecture unique is its simplicity and ability to show the connection between certain filters and specific objects those filters learned in a dataset with purity. The simplicity allows for fewer points of error and easier manipulation of the semantic qualities of generated samples. In practice, DCGANs was able to generate much more realistic images of faces than previous architectures and started to get attention in the media for their application of generating new videos with faces of famous figures known as deepfakes. DCGAN became a backbone used for many more variants being created in the following years, which created a Cambrian explosion of new GAN architectures being created.

One such model was the Wasserstein GAN (WGAN) created by Facebook researchers. It became famous due to its ability to train without requiring balanced training of the generator and discriminator as well as reducing the mode-dropping issue found in previous GAN architectures [68]. In the WGAN, the discriminator is able to be completely trained to its optimal and provides a loss directly to the generator, so as the discriminator increases its accuracy it provides higher quality gradients to train the generator with. As the generator neural network's architectures vary, WGANs were found to be more robust than GANs [68]. The researchers also created a new metric, the Wasserstein estimate, that describes the generator's convergence and quality of the generated samples, which when graphed against the number of training iterations is useful for model debugging and hyperparameter tuning.

Interpretability has always been an important goal when creating and using Deep Learning models, and one such variant of GANs focuses on learning interpretable and meaningful representations of the data through applying feature representational learning [69]. The Information Maximizing Generative Adversarial Network (InfoGAN) is a completely unsupervised learning model that is known for its ease of training, and its interpretable results that prove to be crucial in applications and competitive with representations from supervised learning models. InfoGAN was able to uncover and represent writing styles from digit shapes in the MNIST dataset, poses from 3D rendered images when lit, background digits from the central digit on the SVHN dataset, and hairstyles and emotions on the CelebA face dataset.

Researchers from UC Berkeley found a solution to the image-to-image translation task, where a new image is generated that is a modification of the original image, and created the CycleGAN model. The biggest issue with training a GAN for image translation is that a large dataset of original images and their desired modified versions are extremely expensive to create and often don't exist. CycleGAN instead trains two discriminator and generator models by using cycle consistency to have the first generator learn the original images and generate the modified versions which are then learned by the second generator to generate new images as close to the original as possible [70]. An additional loss metric is a difference between the generated original image from the second generator and the original image which guides the generators toward generating translated images. One of the most impactful applications of this methodology is style transfer where the modified image is of the same style as the original. For example, an image of a dog can be style transferred with a Van Gogh painting to have CycleGAN generate a new image of the same dog as Van Gogh would have painted.

Researchers at Nvidia further utilized style transfer techniques to create the StyleGAN model that automatically learns high-level attributes and stochastic variation in images, the "styles", that allow the generator to create new images with scale-specific control with better interpolation properties and better latent variable disentanglement [71]. In practice, this allows for the significantly higher quality and more realistic generation of human faces as the "styles" learned by the generator are the unique characteristics of the images of human faces from the training dataset.

Researchers from Google created the famous Transformer architecture that connects the encoder, the generator, and the decoder, the discriminator, through an attention mechanism [72]. An attention mechanism looks for the most relevant parts of a dataset through using a weighted combination of input vectors with the most relevant input vectors granted the highest weights. This same architecture was then applied to a convolutional based GAN model called the Self-Attention Generative Adversarial Network (SAGAN) [73]. SAGANs are able to generate data with details across the entire high resolution feature map as opposed to previous models only generating data with details in local points from lower resolution representations of the data. With SAGANs being attention-driven from utilizing the self attention architecture in both the generator and discriminator, they are able to converge must faster than other GAN architectures and achieve state of the art results with the ImageNet dataset [73].

Ultimately, all of these GAN architectures mentioned show how fast the field of Deep Learning has been developing in recent years, and how the vast usecases of GANs have allowed them to become one of the most impactful Deep Learning models in recent times.

4.1.2. Mathematics

We define the Discriminator as function D(x), which determines the probability of an input x being a real piece of data as opposed to one generated by the Generator. We also define the Generator as function G(z), which generates a piece of data based on input z.

We train the function G(z) to fool the Discriminator by training it to minimize the value of log(1 - D(G(z))). Note that as D(G(z)) gets larger

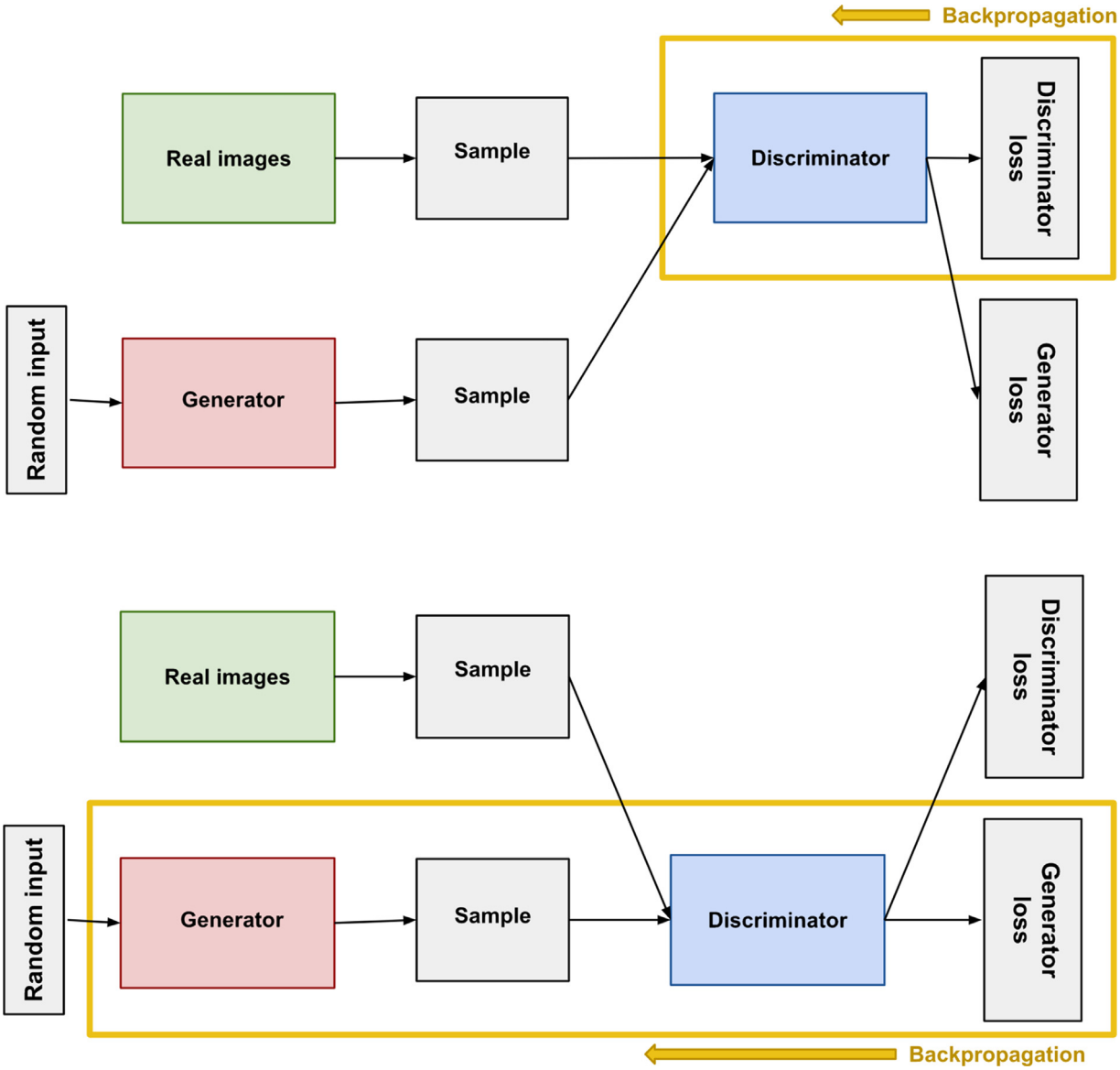

Fig. 3. Backpropagation in Discriminator and Generator Training [11].

(indicating that the discriminator believes a piece of generated data is real), log(1 - D(G(z))) gets smaller. Simultaneously, we train the function D(x) to assign the correct labels to both the real data and the samples from G.

Therefore, the loss function of a generative adversarial network can be modeled as the following:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z))] \quad (9)$$

The interplay between the Generator and Discriminator can be modeled mathematically as a zero-sum game, a situation where one party's improvement is always equal to the other's loss - in other words, the gains and losses of each party always sums to zero.

For example, imagine that the Generator creates images that are classified as real by the Discriminator 25% of the time. Therefore, the Generator has a 25% accuracy, and the Discriminator has a 75% accuracy. Now imagine that the Generator improves itself with feedback from the Discriminator, so that the Generator is now 80% accurate. Therefore, the Discriminator must be 20% accurate. In other words, the Generator has improved by 55%, and therefore, the Discriminator must have lost 55% points of accuracy.

However, this poses a problem for creating a loss function for a GAN. If we try to only maximize the performance of the Generator, the best course of action would be to have the Discriminator perform poorly, and vice versa. In 4.1.1, future variants of GANs discussed all tweaked their equations to minimize this situation from occurring (Table 1).

5. Applications of GANs

General Adversarial Networks have found a range of applications, helping to overturn and accelerate innovation in industries ranging from video prediction to realistic photograph generation and medicine.

For our review we focus on the drug discovery and development based applications of GANs [44,74–78].

5.1. Drug discovery

Drug discovery is the process in which new medications are discovered. Traditionally, drug discovery worked in a very counter-intuitive way to what we see today. Chemists would study the effects of different remedies without a biological target in mind, meaning they hypothesized some chemicals would have effects and they tested to see where in

Table 1
Summary of Popular GAN Variants.

CRITERIA	GAN	CGAN	LAPGAN	DCGAN	AAE	INFOGAN
Learning	Supervised	Supervised	Unsupervised	Unsupervised	Supervised, semi-supervised and unsupervised	Unsupervised
Network Architecture	Multilayer perceptrons	Multilayer perceptrons	Laplacian pyramid of convolutional networks	Convolutional networks with constraints	Autoencoders	Multilayer perceptrons
Gradient updates	SGD with k steps for D and 1 step for G	SGD with k steps for D and 1 step for G	No updates	SGD with Adam optimizer for both G and D	SGD with reconstruction and regularization steps	SGD updates to both G and D
Methodology	Minimize value function for G and maximize for D	Minimize value function for G and maximize for D conditioned on extra information	Generation of images in coarse-to-fine fashion	Learn hierarchy of representations from object parts to scenes in both G and D	Inference by matching posterior of hidden code vector of autoencoder with prior distribution	Learn disentangled representations by maximizing mutual information
Performance metrics	Log-likelihood	Log-likelihood	Log-likelihood and human evaluation	Accuracy and error rate	Log-likelihood and error-rate	Information metric and representation learning

the body the effects would target. This style of medical practice has now become outdated in the modern world as different technologies such as computing has allowed researchers to gain knowledge on how to identify on specific targets in the body in order to study their effectiveness. It constitutes computational and experimental research that is undergone in order to identify potential new drugs to address certain medical conditions. The process of drug discovery is an arduous and resource-intensive one [102], and GANs may have applications in streamlining it [94,103–106].

An area that makes GANs an exceptionally strong tool in drug research is how it can be used in searching for new molecules. An issue with the traditional method of drug discovery involved an inability to exhaustively cover all possible outcomes in a chemical space, which is a concept in chemistry referring to the property space of all chemical compounds bound by a given set of conditions and principles. What this means is there are so many possible molecules that it is not feasible for a researcher to find the best options when looking between similar compounds which might have comparable, yet slightly contrasting, effects as a drug. It should be noted that GANs are not perfect in the field of chemical space, but they are more successful here than any competing method. This is where GANs strive, as deep learning models can be constructed to help find compounds with desired functionality [107,108]. Different biological models such as evolutionary data, diversity algorithms, and population trends can be used to create training data for the GANs. Once the GAN is and trained on those types of data, two main results are produced. Firstly, there are several new drug variations which can then be tested for their functionality and have their side effects minimized. Secondly, this production allows the GANs to expand or update its generated variations and constantly improve within the constraints of the chemical effect the researcher is seeking during the training process. However, an issue with this method is mode collapse, which causes the generator to only produce samples closely related to a tiny subset of its training data. To combat this, a study done by Blanchard et al constantly updated the training data and recombined during replacement [107]. This minimization of the core weakness for the use of GANs in this area shows massive future potential for the future in accelerating drug discovery (Table 2).

Table 2
Summary of Most Popular Applications of GANs.

Study	Application
AGEGAN [3]	Change facial attributes
DR-GAN [79]	Change facial attributes
SD-GAN [26]	Change facial attributes
SL-GAN [80]	Change facial attributes
VAE-GAN [81]	Change facial attributes
CAAE [82]	Change facial attributes
Age-cGAN [3]	Change facial attributes
DCGAN [30]	De-occlusion
U-nets and GANs [32]	De-occlusion
CyCADA [42]	Domain adaptation
FD-GAN [33]	Human Pose identification
GP GAN [83]	Image blending
CycleGAN [84]	Image translation
DiscoGAN [85]	Image translation
PAN [86]	Image translation
Pix2pix [77]	Image translation
Coupled GAN [87]	Joint image generation
DI2IN [88]	Medical image segmentation
SCAN [88]	Medical image segmentation
SegAN [89]	Medical image segmentation
C-RNN-GAN [90]	Music generation
ORGAN [37]	Music generation
SeqGAN [91]	Music generation
Perceptual GAN [92]	Object detection
SeGAN [28]	Object detection
Perceptual GAN [93]	Object detection
Anime GAN [94]	Object detection
GeneGAN [95]	Object transfiguration
GP-GAN [83]	Object transfiguration
PN-GAN [96]	Person re-identification
IPGAN [97]	Person re-identification
VAW-GAN [44]	Speech conversion
SRGAN [98]	Super resolution
WGAN [4]	Super resolution
Stack GAN [74]	Text to image
TAC-GAN [22]	Text to image
MoCoGAN [99]	Video generation
Pose-GAN [100]	Video generation
VGAN [101]	Video generation

Two of the issues in relation to drug research that GANs are being used to help resolve are the issues involving the heavy cost and overall time for drug research, but one issue that has come up for GANs itself involves the limitation when working to explore some regions of chemical space. We previously referenced how GANs work better than other traditional methods as well as newer deep learning when it comes to chemical space, but innovations in GANs look to expand and solve this shortcoming of not having complete access to chemical space. A fully quantum GAN could accelerate the training process for GANs, as well as offer better training samples which could allow more exploration of chemical space but lacks the ability to process over two dimensions [109]. In the future, a fully quantum GAN could be viable and successful, but more innovation in the technology is currently needed. In the meantime, a paper proposing a qubit-efficient quantum GAN mechanism with a hybrid generator and classical discriminator as a tool in drug research. Due to requiring less qubits, complex simulation would still be the main function of this model while requiring around 20% less training time and being more successful in searching chemical space [109]. Mode collapse is again an issue with this model. Even with the swifter training, mode collapse can still occur meaning the GAN learns one aspect of the data quite well, but not all the data is being properly learned causing incomplete training. The results of this model showed a 98% reduction in generator parameters and created the expected benefits of the hybrid system with quicker data learning as well as more success-

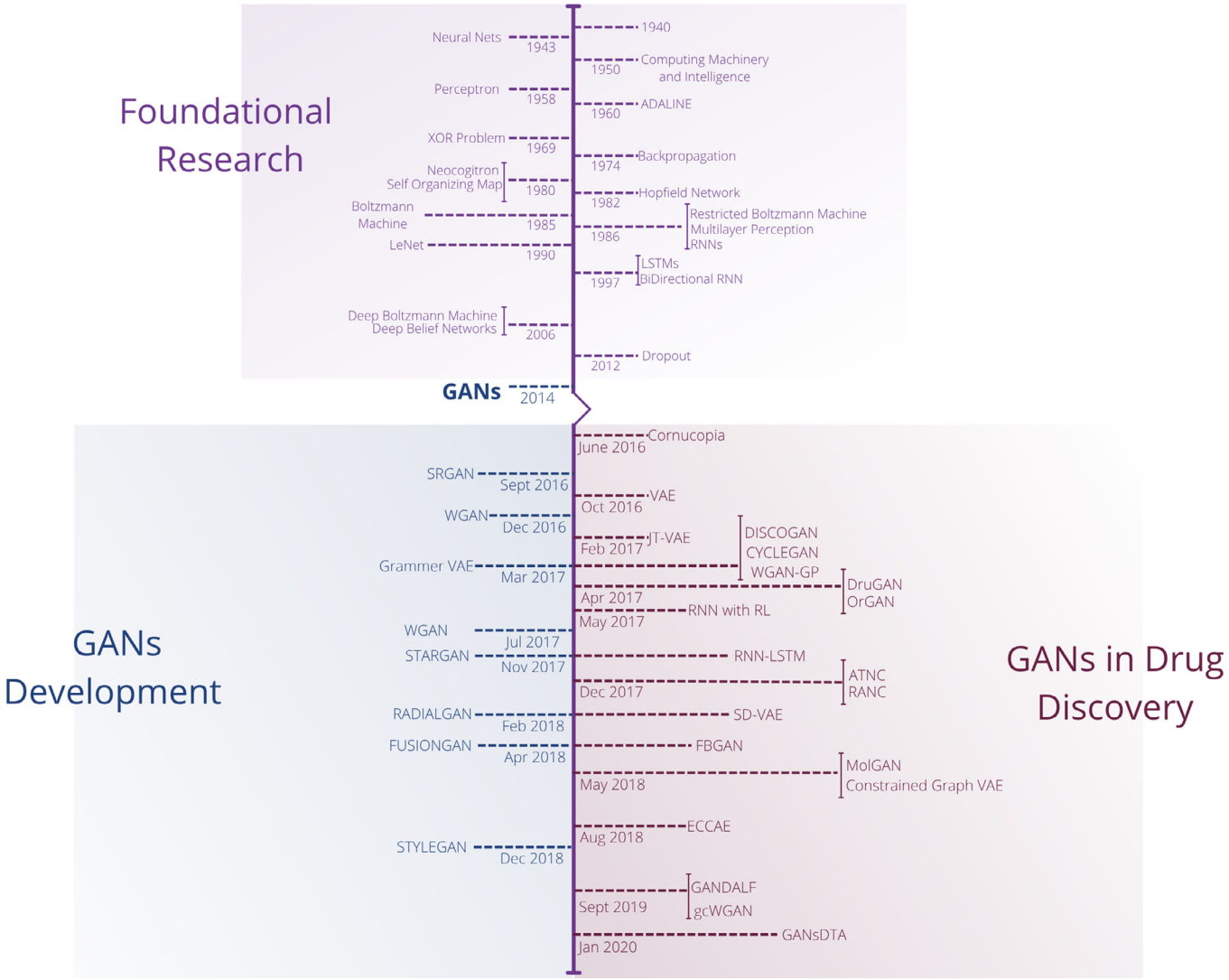

Fig. 4. A Timeline Depicting the Major Advances Leading Up to GANs and Development in Drug Discover.

ful chemical searching. The reduction of parameters allows for much swifter training and simpler methods of hyperparameter tuning (Fig. 4).

GANs are being seen to have applications in *de novo* peptide and protein design, a process through which new peptides and proteins are discovered. This process entails generating new chemical entities that fit a set of constraints using computational algorithms. Hence, the chemical entities can be generated without a starting template from the beginning, reflecting the meaning of the phrase *de novo* [110]. As summarized by Lin et al, [109], there have been many studies done regarding the applications of GAN-based approaches in *de novo* peptide and protein design. Sabban and Markovsky [111] proposed that the LSTM-GAN (Long Short-Term Memory Generative Adversarial Network) structure was able to generate new helical protein backbone topologies. While the LSTM-GAN structure had originally been devised for applications in Natural Language Processing, this group of researchers was able to find uses for it in generating protein backbone topologies with features beneficial for protein design.

Lin et al continue to describe another group of researchers, Karimi et al. [112] who investigated the gcWGAN (guided conditional Wasserstein General Adversarial Network) structure and its applications in peptide folding. This conditional Wasserstein GAN structure is derived from the Wasserstein GAN [68] with gradient penalty. The Wasserstein GAN, as proposed by Arjovsky et al., is a type of GAN architecture which uses the Earth-Mover distance rather than the Jensen-Shannon divergence of a traditional GAN. gcWGAN builds upon a conditional Wasserstein GAN using an oracle. The conditional GAN generates sequences, and the oracle predicts a fold for each sequence. Each predicted fold serves as feedback for the generative module. Furthermore, Lin et al. outline two separate groups of researchers who looked into the DCGAN(Deep Convolutional Generative Adversarial Network) structure and its applications in drug discovery. One group, Anand and Huang [113] found that it can be applied to generate new protein structures. Both the generative and discriminative modules were implemented using a convolutional network structure, a structure traditionally used in computer vision tasks. Another group, Bian et al. [114] proposed that the DCGAN could be used to generate target-specific compounds for cannabinoids through the use of developed convolutional neural network frameworks such as LeNet-5. Thus, both groups of researchers found applications of the DCGAN structure in *de novo* peptide and protein design in drug design through the implementation of the generative and discriminative modules using a convolutional neural network structure (Fig. 5).

Another example of a GAN-based structure's relevance to drug design is in the promulgation of the GANDALF(Generative Adversarial Network Drug-tArget Ligand Fructifier) framework. This framework, implemented by Rossetto and Zhou [115], was able to generate a peptide highly similar to FDA approved drugs for a given target. The GANDALF

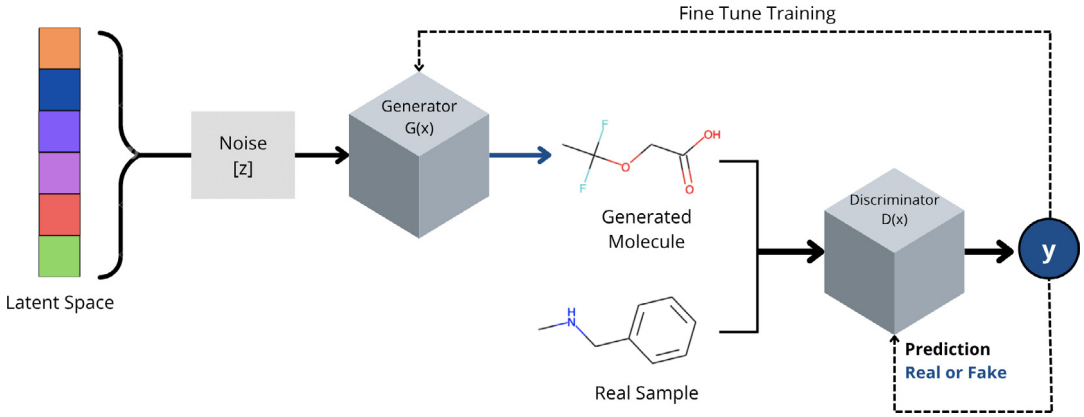

Fig. 5. An Example of Vanilla GAN Architecture for Molecule Development.

framework also uses the DCGAN structure, using a five-layer convolutional neural network for the generative and discriminative network modules.

Another GAN structure, Feedback-GAN, was shown by Gupta and Zou [116] to be able to generate antimicrobial peptides with desirable properties. This structure consists of a GAN and differentiable neural network analyzer. This differentiable neural network analyzer is a prediction algorithm that determines if a gene sequence is able to encode a peptide. The analyzer and the GAN architecture are connected through a feedback-loop training mechanism which allows the generative module to create valid sequences. In each epoch, the generative module produces many sequences, which then receive a score from the differentiable analyzer. The sequence with the highest score is then passed as input to the discriminative module. The discriminative module receives both real inputs and fake inputs(in the form of the generated sequences) then differentiates between the two types of inputs.

A research used GANS with adaptive training data for drug discover. During the training period, the model saved new and valid compounds it generates. The training data then updated using a replacement technique that may be either directed or random. The technique was repeated when the training resumes. The findings suggest that this method can counteract the decline in new molecules created by a normal GAN during training. In addition, using recombination between created compounds and training data to boost novel molecule discovery. By including replacement and recombination into the training process, GANs may be used for larger drug discovery searches[107].

Researchers have also found applications of GANS in regards to molecular *de novo* design as well as *de novo* protein and peptide design. Guimaraes et al. [117] proposed the ORGAN(Objective-Reinforced Generative Adversarial Networks) structure, a structure combining the GAN with reinforcement learning. More specifically, the structure consists of a Sequence GAN model and the recurrent neural network model. The discriminative module is implemented as the convolutional neural network model and the generative module is the recurrent neural network model. Guimaraes et al. found that the ORGAN structure was sufficiently able to generate new molecular compounds with preferred properties through a drug development pipeline. Furthermore, the ORGAN structure was able to do so with better results than just a recurrent neural network model or a GAN model alone.

5.2. Drug development

With the traditional method of drug discovery, it was said how it previously was difficult to specifically target areas in the body in research, which prompted researchers to work backwards in testing remedies with previously unknown effects. Along with other modern methods, this is a key area in which GANs look to improve research in drug discovery. GANs can be used to predict binding affinity, and therefore the successful of a drug being researched [118]. Deep learning models can work to identify drugs which have high affinity when binding to specific disease proteins, which is beneficial for drug discovery as it expedites the process for researchers. The strength of GANs in this area comes from advantages in both speed and cost. Other machine learning models in the past have been able to accomplish similar predictive ability, but GANs require less manual input as well as need less expensive costs for building their training data due to the reusing of parts of the generator and discriminator networks as feature extractors.

A study on maximizing the effectiveness of GANs in this area used three elements: a feature extractor for protein compound, a feature extractor for protein sequence, and a regressor for affinity value prediction [118]. In the first round of training, the fake samples are generated by the generator of the GAN and both the fake as well as real samples are put into the discriminator. The discriminator will then work to classify the samples as real or fake so that the generator can create more realistic protein sequences and compounds, which are labeled by the discriminator. In the second round, labeled protein sequences are used to train the regressor to minimize its loss and create optimal model parameters. These parameters can then be used in a feature extracting model and a regression model [118]. Results from using these models showed that GANs can perform as well if not better than previously used learning-based models with the added factor of GANs training on freely unlabeled data and coming at a lower cost by guiding biological experiments. These newly created models by the GAN help create the beneficial protein sequences which can help biologists in targeting the needed cell receptors in drug discovery. The main downside in results was GANs poor performance when dealing with very small data sets, but this can be offset by regularization techniques to train GANs.

5.3. Biomolecular

GANs are being used to create synthetic DNA sequences that code for proteins of varying lengths. One study used the feedback-loop process to generate synthetic genes that code for antimicrobial peptides and to optimize synthetic genes for the secondary structure of the peptides they produce. A number of measures show that the proteins created by GAN have desired physicochemical characteristics [116].

DL models are used to learn the nonlinear probability distribution between molecular chemical structures and their biological and pharmacological characteristics from enormous data sets, and then execute in silico creation of *de novo* molecules with desired features in the laboratory [120–122].

For example, a QSAR methodology is defined by an ML application and/or statistical approach to uncover empirical connections where the biological activity (or any other attribute of interest) for molecules is stated in terms of estimated molecular descriptors.

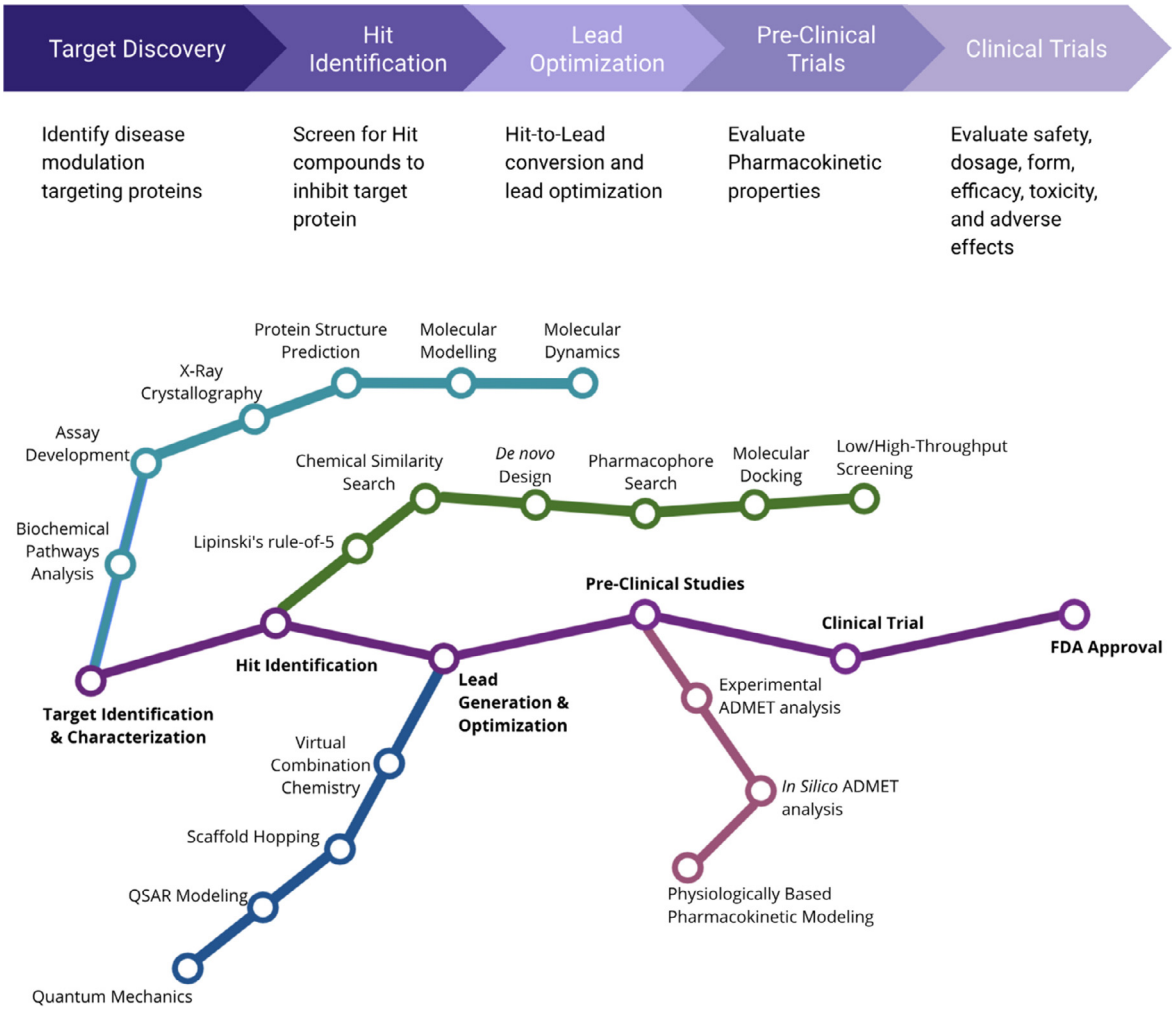

Fig. 6. Several successful uses of machine learning in various phases of the drug development pipeline in pharmaceutical companies have been documented. These applications may be found in a number of different places. A conceptual map on the experimental and computational methodologies that are applied to the drug discovery process [119], together with a schematic explanation of the drug discovery process that is overlaid with the related computational approaches. There is no particular hierarchy to the order in which the terminologies are listed on any of the colored tracks.

SBDND and LBDND are the two most extensively used computational methodologies for designing new drugs based on structural and ligand-based principles, respectively. X-ray structures or relatively realistic homology models of the intended targets are required for SBDND in order to create ligands, which are then assembled from atoms or fragments in order to suit the necessary pocket [123,124].

DruGAN employs fingerprints to describe molecules, and utilizes Tanimoto resemblance to assess how similar the created molecules are to the original data, In the end, DruGAN had a superior ability to generate chemical fingerprints and a greater capability to analyse extremely huge data sets of molecules than its competitors.

It's also worth noting that VAE and AAE work well with fingerprints and other chemical structure representations as well. Various forms of generative adversarial autoencoder models were used in inverse QSAR to build new chemical structures (Figs. 6, 7, 8).

5.4. Targeting

The current issue with systemic drug administration is the need for a high concentration of a drug to achieve a therapeutic concentration in the site of action. However, the use of high concentrations can cause unnecessary adverse side effects. To combat this, drug targeting aims to target and guide the administration of drugs to the target area which would resolve the high drug concentration. There are many techniques to accomplish this including rudimentary methods such as direct application to the affected area, or more complex techniques utilizing normal pH values, temperatures, or magnetic fields. Yet, the most versatile drug targeting technique is achieved by using a unique vector molecule. Vector molecules are ligands that have an increased affinity towards an area of interest. The potential benefits include highly configurable drug delivery, which allows for higher loading capacity, customizable size, and customizable permeability [125].

Utilizing computational predictions of drug targeting interactions (DTI) is a challenge today. Currently, DTIs are primarily based upon supervised machine learning with known label information. This creates an expensive and time-consuming process. One solution is a semi-supervised generative adversarial network (GANs) based method to predict binding affinity. By the use of computational methods and machine learning, the drug development process can be considerably accelerated and save excessive costs [118].

The model proposed in this paper, GANs DTA, contains three parts: two feature extractors and a regressor [118]. The first feature extractor is for the protein sequences while the second one is for SMILES strings, which is a simplified molecular input line entry system. These strings

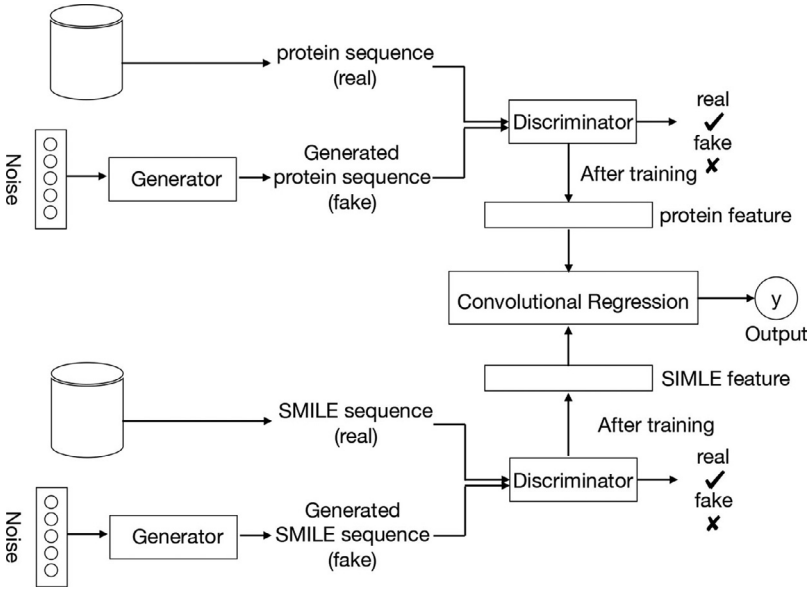

Fig. 7. Zhao et al's pipeline involving two GANs, one for generating protein sequences and the other for SMILE sequences [118].

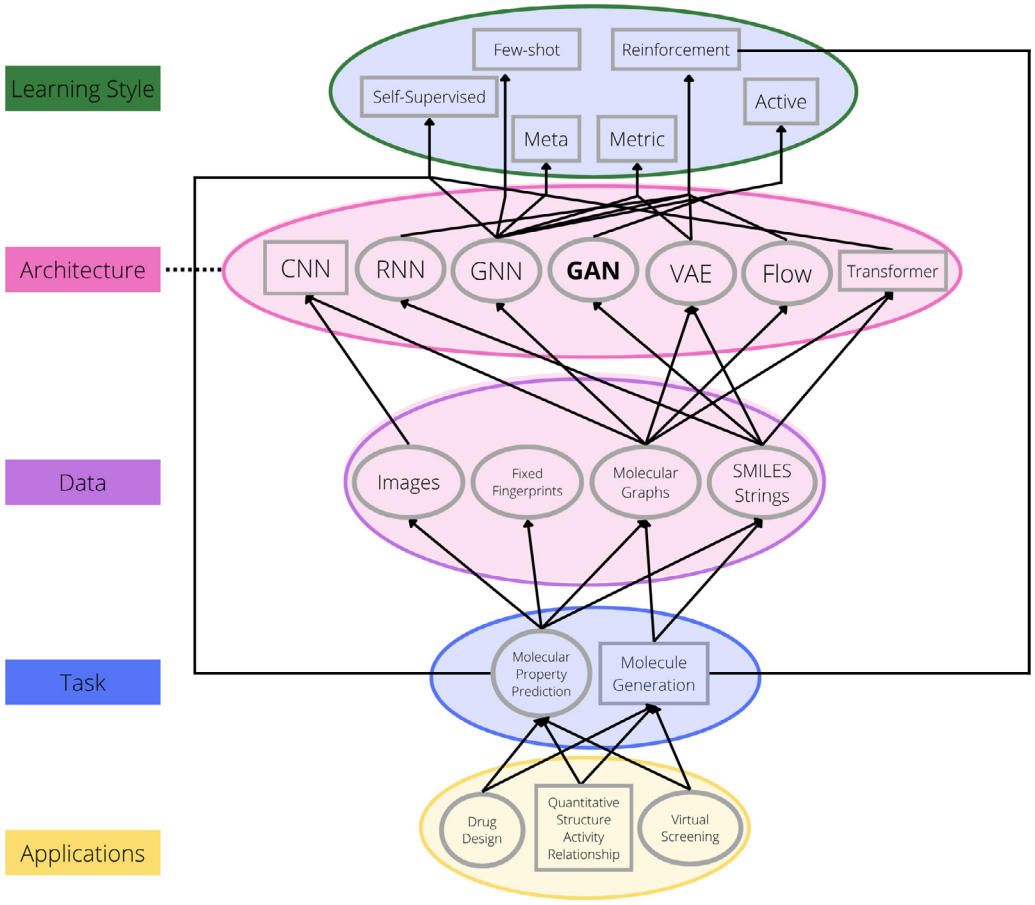

Fig. 8. A graph based analysis to represent the relation and application of different deep learning techniques to molecular data and tasks.

are made of a feature representation modular from GANs. The third part is a regressor, made up of a Convolutional Neural Network (CNN), which predicts the affinity value. To train this model, it must undergo two processes. The first of which is training the feature extractors in the context of GANs. The GAN begins to generate fake samples starting with a given noise distribution. It is then compared to a real provided sample to discriminate the real data from the fake generated data. As this continues, the discriminator maps the inputs utilizing a feature space by a local feature extractor. This local feature extractor promotes sample classification, which learns the difference between real and fake proteins and SIMILES. After the first round is completed, the local feature extractor will be used as the feature representation of the new model. The second round is to build upon the regressor. This uses the feature extractor from the first round along with the labeled data to train the

regressor to minimize the last function, leading to optimal model parameters. Now, this can be used with drugs that are represented as SMILES strings and protein sequences that are represented as a string of ASCII letters, which are amino acids [118]. Allowing for the input to be in the form of a string allows for the discriminator to learn the latent features of those sequences. Ian Goodfellow et al suggest a minimax game to train GANs [64]. The generator G creates fake samples from the generator distribution P_G by transforming the noise variable $z \sim P_{noise}(z)$ into a sample $G(z)$. P_G is then compared to the real sample distribution P_{data}. Equation (9) describes the loss function that is iteratively run when training a GAN [118].

The discriminator network used in the drug targeting affinity architecture consists of the feature extractor ($F(x; \phi f)$) and a sigmoid classification layer with weight vector ψ_1 [118].

The convolutional regression model conducts convolution operations with kernel size 4 to create feature maps of the input data. The dimensions of the convolution layers are 16x4, 32x4, and 48x4. The output layer is a linear function that obtains the continuous value. This is on a network that is trained to decrease the loss of function defined by the mean square error (MSE) between the depth values y_k and the predicted outputs of the network p_k.

$$MSE(p_k, y_k) = \frac{1}{n}\sum_{k=1}^{n}(p_k - y_k)^2 \tag{10}$$

The results of this experiment were compared to state-of-the-art drug-target binding affinity prediction models using the Davis and Kiba data sets, eighty percent of which were used as training samples whereas the other 20 percent were used for testing. The MSE and the concordance index (CI) were later compared to evaluate the models [118]. CI evaluates the performance of the models that output continuous values. In both data sets, Davis and Kiba, DeepDTA outperforms with the best CI (0.886 and 0.863) and the best MSE (0.261 and 0.194). The best performance GANsDTA exhibited was 0.866 for the second dataset CI. Overall, GANsDTA exhibits similarities in performance to DeepDTA, however, has slightly a lower CI score (0.881 to 0.886) and a slightly higher MSE with 0.015. The likely conclusion is the insufficient size of a data set that only included 442 proteins, 68 compounds, and 30,056 interactions were too limited [118]. Although this model is not meant for small data sets, employing the techniques of training GANs should be continued to enhance the adaptability of GANs on a larger scale.

The endocannabinoid system is made up of two main subtypes: CB1 and CB2 receptors. The first is commonly found in the central nervous system while the second is mainly concentrated in hematopoietic cells and the immune system. It has been proven that the first type of receptor shows the potential to treat anxiety, drug addiction, and even motor control, while the second type of receptor has the potential to alleviate inflammatory pain, osteoporosis, and autoimmune disorders. This being said, the cannabinoid system can be difficult to target when it is dispersed throughout the whole body. This is the reason why developing small molecular drugs capable of targeting types of CB receptors is essential for the medical treatment of various diseases [114] (Figs. 9 and 10).

In order to target these cannabinoid receptors, a study utilized a deep convolutional generative adversarial Network (DCGAN) model for *de novo* designing target-specific compounds [114]. *de novo* is one of two strategies for identifying the compounds of a specific target. In this study, four types of targets, or fingerprints, were used to describe the small molecules with different structural features: ECFP6, MACCS, AtomPair Count, and AtomPair. A convolutional neural network (CNN) was the architecture for the deep learning model used in this study. It was first built using the four aforementioned types of targets as input data. Then it was applied to the discriminator of the DCGAN. Afterward, the generator was created through reverse convolution. This strategy minimizes the discriminative and generative loss simultaneously [114].

The label data used was from their previous machine learning project on cannabinoid receptors. The activity cut-off was set to 100 nM to distinguish active from inactive compounds. The cut-off value was set to 0.8 since no pair of compounds had a cut-off larger than that. Then 80% of the data set was used for training while the latter 20% was used for testing [114]. This was split between the four fingerprints. The dcGAN had a composition of two deep neural networks, the generator G and the discriminator D. As G undergoes reverse convolutional processes to capture the data of distribution resulting in the fake molecular fingerprints, D is trained to discriminate the data, using the preferred CNN architecture to detect whether it is real or not [114].

Training the model is a two-step procedure. The first step is training the discriminator to minimize the discriminative laws based on the active ligand fingerprints. G then creates fingerprints based on the noise input with the label of zero. The discriminator chooses between these fingerprints and the authentic data which is labeled as one. The second step is minimizing the generative loss. Here the discriminator is not trainable and the generative loss is recorded to show how well the generator fools the discriminator. The learning epochs were 50, 100, and 200, while the batch size was 128. When comparing the different CNN architectures, the LeNet-5-based architecture and the Atompair fingerprints performed the best. Based on the ROC curves, this duo attained the highest AUC score across each test set for both types of receptors, CB1 and CB2 [114]. This is the optimum combination using the first as the generation for the discriminator and the second as the input data. This strategy lets the machine produce fingerprints of a potential target-specific compound without any effort from the user.

The study does point out two concerns regarding the GAN architecture. The first challenge is to optimize both the discriminator and the generator simultaneously. However, this instability can lead to a gradient favoring one over the other. This could result in an improved score for D, but not G, or vice versa. Secondly, there can only be a restrictive set of outcomes to be generated, also known as mode collapse. With the restrictive set, the generator can only produce one type of outcome or a small sample. Only a limited chemical space is to be covered as well as a lack of structural diversity.

In 2019, people realized how time consuming and expensive the traditional drug discovery process was. On average, it takes about ten to fifteen years and billions of US dollars for a drug to reach the FDA, at which point it may be accepted or rejected. In drug discovery, drugs are created to find two specific proteins, which gives rise to the importance of a computational drug discovery system to protect the binding affinity, also known as drug-target interaction prediction (DTI). The binding affinity can be represented with multiple terms, some being the inhibition constant, the dissociation constant, changes in free energy measures, half maximal inhibitory constant, half maximal activity concentration, KIBA score, and SCORES used in STITCH database [126].

This study proposes a DeepGLSTM model, using a graph convolutional network block to pass the drug compound data using power graph representation and bidirectional-LSTM to capture the topological information to achieve state-of-the-art results in the neural drug repurposing domain [126]. As many other papers have focused on one-dimensional CNN models, this model is LSTM since existing literature proves it performs better. Most GAN models convert drug compound data into string representations, which is not ideal for computational purposes. It may lead to the exclusion of topological information of the molecules themselves, which causes a decrease in performance and power production.

Deep-GLSTM consists of two functional models. One captures the topological information from the drug molecules while the second one captures the sequential information from the specific protein structure. It has been shown through previous papers that Graph DTA and DeepGS using graphical representation to embed the drug compounds and one-hot representation for protein structures improved the model performance and prediction [126]. This study continued with this idea by representing each drug compound by it's related SMILES notation. This notation represents a compound as a graph of the interaction between each atom. A node feature was a set of atomic features adapted from DeepChem. A node represented a multi-dimensional binary feature vec-

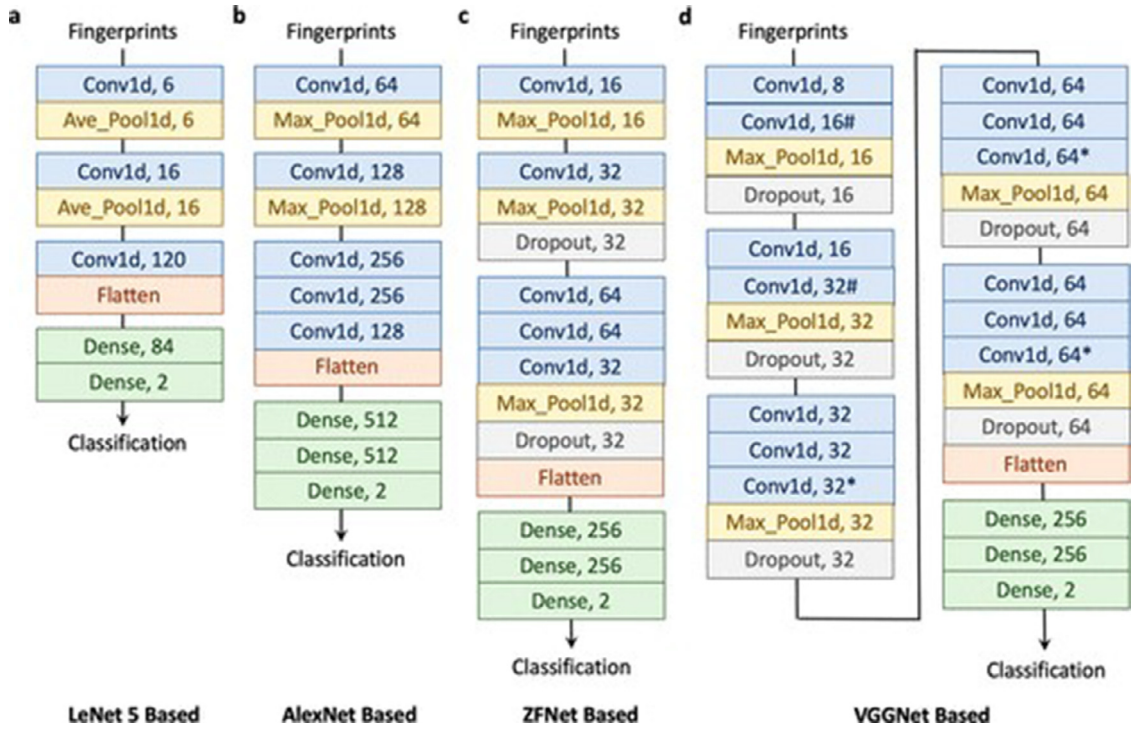

Fig. 9. Modified and Applied CNN architectures used as the GAN generators and discriminators [114].

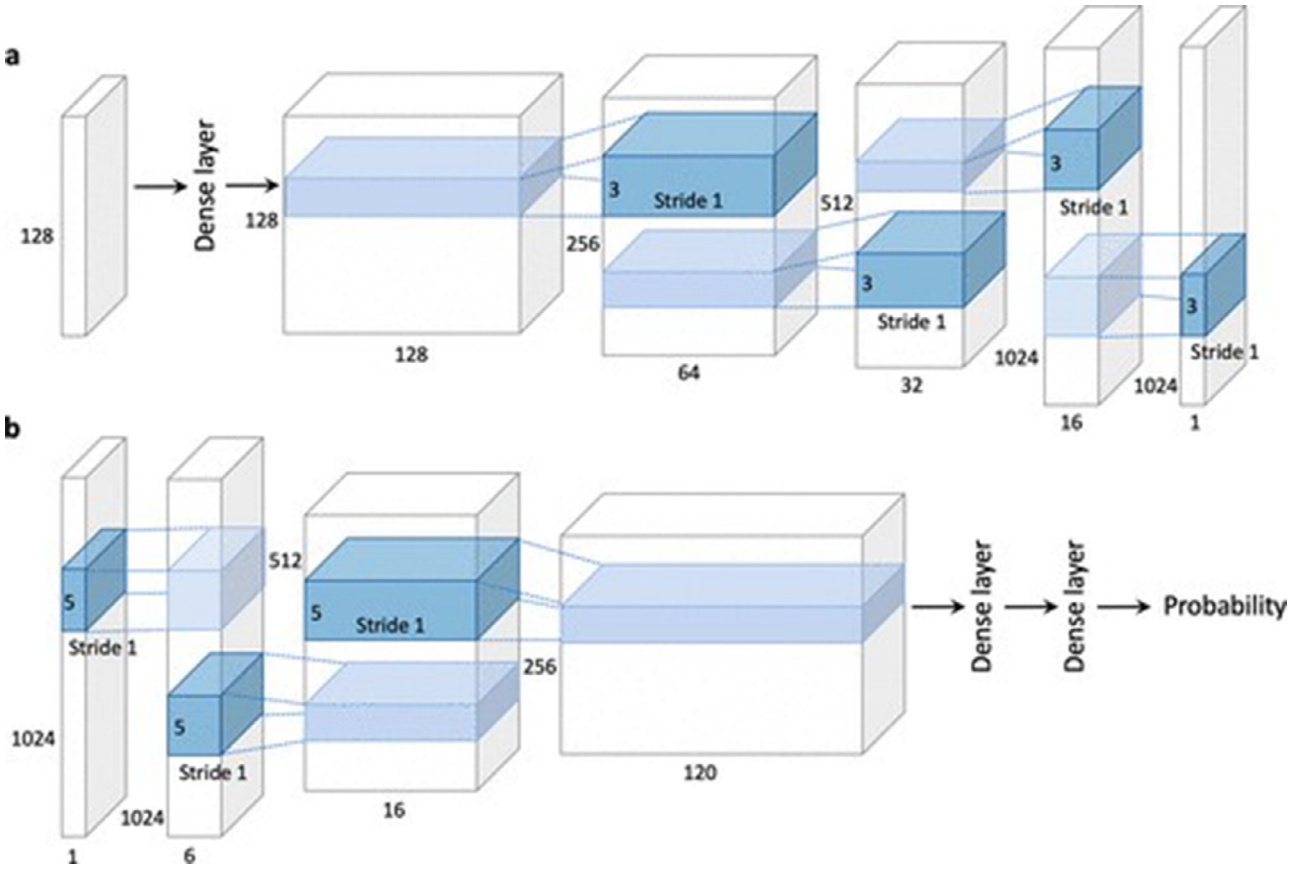

Fig. 10. DCGAN generator and discriminator used for molecular compound fingerprint generation [114].

tor. Each feature vector demonstrated five pieces of information including the atom symbol, the number of adjacent atoms, the implicit valence of the atom, the number of adjacent hydrogen's, and whether that item is an aromatic structure. The SMILES data was later converted to a molecular graph representation which helped extract the required atomic features.

Every amino acid was mapped to the english letter based off the protein sequence being a string of ASCII characters. Then the embedded representation, where d_p is the dimension of the protein sequence embeddings to a Bi-LSTM layer, captures the relationship of the characters $p = [c_1, c_2, \ldots c_n]$ of length n. The prediction of the affinity score was based off of the interaction of each node with its adjacent node [126]. This is why a GCN was used, due to the fact that it is strong enough to produce important feature representations, which are capable of capturing the connectedness relationship between nodes of the graph. This model consisted of three blocks. The first had a stack of three GCN layers. The second had a stack of two GCN layers. The final block had a single GCN layer. The first equation is the overcome degree normalization of the adjacent representation. The output is the normalized adjacency representation.

$$A_{norm} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \tag{11}$$

The next equation makes the first block workable. W is the trainable weight, $H_e^i = X$ is the ith layer output represented by the sigmoid as a non linear activation function:

$$H_e^i = \sigma(A_{norm} H_e^{(i-1)} W^{(i-1)}) \tag{12}$$

The second block is the square representation of the first equation. The same thing can be said for the equation that makes it workable.

$$A_{norm}^2 = D'^{-\frac{1}{2}} A^2 D'^{-\frac{1}{2}} \tag{13}$$

$$H_e'^i = \sigma(A_{norm}^2 H_e'^{(i-1)} W'^{(i-1)}) \tag{14}$$

Similarly the same can be said for the third block as seen in the above equations (Fig. 11).

The length of the protein sequence was set to one thousand for all datasets. The hidden vector representation of each layer was one hundred and twenty eight while the fixed number input mode features was seventy eight. The final hidden vector representation of the first block with set the three hundred and twelve. The dropout probability rate was set to 0.2. Due to the fact of it being a regression model, the mean squared error (MSE) was used as the loss function evaluation. Finally, all experiments were trained to one thousand epochs. The results suggested that the third block of the GCN contributed the most in model performance [126]. Most often, the structure of a drug dose does not consist of many shortest paths that exceed a distance of three or more, therefore it can be concluded that increasing the exponent to be on 3 would not lead to any significant changes.

6. Outlook

The application of ML methods and the latest advancements in DL offers a multitude of potential to boost productivity throughout the drug research and development pipeline. As a consequence of this, in the coming years, we anticipate seeing an increase in the number of applications within the industry that target clearly defined challenges. ML algorithms are going to systematically generate improved outputs, and new and interesting applications are expected to follow suit as a result of this. This is because the available data is getting "bigger," at least in the sense of covering the relevant variability of the entire data space in a more comprehensive manner, and because computers are becoming increasingly more powerful. This has been demonstrated quite clearly in the sections that came before it. In those sections, we talked about some machine learning applications for target identification and validation, drug design and development, biomarker identification, and pathology for disease diagnosis and therapy prognosis in the clinic (Tables 3 and 4).

These strategies are also being implemented in the healthcare sector, where they have the potential to significantly improve precision and personalized medicine [161] when combined with the drug development process. To further improve clinical trial outcomes and the process of deciding if a patient is qualified for a study, ML has been applied to Omics and EHR data [162] as well as other real-world biomarkers. A recent study, for instance, showed that deep neural networks (DNNs) are a highly competitive way for autonomously collecting useful information from electronic health data for the goal of disease detection and classification [163]. For prognosis prediction, research has shown that machine learning models embedded in EHRs can outperform more traditional approaches. Applying machine learning to the data now being created by sensors and wearables can help us better understand the disease and develop treatments, especially in neurosciences [164]. For example, Gkotsis et al. [165] classified mental health disorders using unstructured data from social media using DL algorithms. This is a difficult problem for standard ML methods to solve.

Lack of interpretability, or the inability to provide a credible answer of how the trained neural network comes to the outcome, is a common problem with deep-trained neural networks. Even in cases where neural networks outperform human specialists, a lack of interpretability could pose problems for researchers, regulators, doctors, and patients if the system is used to detect a condition based on medical imagery, like melanoma, for example. Can we expect patients to place more faith in the ML diagnosis than they would in a human doctor's opinion? Much less dramatically, a comparable problem may arise when designing drugs. Would a pharmaceutical company put its faith in a neural network to select a tiny chemical for its portfolio and fund its advancement in the clinic if the network couldn't explain its reasoning behind the selection? Further, if chemicals have been developed by computer algorithms, there may be inventorship concerns while applying for patents. In any event, researchers should treat ML results as only hypotheses or intriguing jumping-off points for further investigation. Though complementary tests validating the ML result will aid in establishing confidence in methodologies and outputs, regulatory bodies have not yet provided guidance on how they feel about ML's lack of interpretability in clinical settings. The lack of interpretability of the approaches, however, makes it harder to debug these methods when they fail in unexpected ways on fresh, unexplored data sets, even if the trust issue is resolved.

Because ML outputs are largely dependent on the initial values or weights of the network parameters or even the order in which training examples are presented to the network, all of which are normally chosen at random, repeatability is another key problem for neural networks. Should we expect the network to consistently choose the same illness target when fed the same expression data? Would the medication structure always be the same that ML suggested? Various techniques have produced different prognostic biomarkers for breast cancer based on molecular expression characteristics [166], illustrating that this lack of repeatability is particularly troublesome for biomarker identification. The fact that several ML approaches can provide divergent outcomes raises doubts about the widespread use of these techniques. It has been suggested that there are ways to address both the interpretability and repeatability issues. It is common for these to include employing a more involved or time-consuming technique or averaging the output of multiple network models, although this may be viewed as doing little more than adding yet another possible outcome to the pool (Tables 5 and 6).

Large amounts of high-quality, accurate, and human-curated data for training and developing ML models are also vital to consider. The complexity of the data type and the question to be answered determines the requirements for the amounts and accuracy sought. As a result, the cost of creating these data sets can add up quickly. Possible approaches to satisfying these data needs include pre-competitive consortia of pharmaceutical corporations and academic institutions employing suitable data standards and equipped with the requisite operational and open

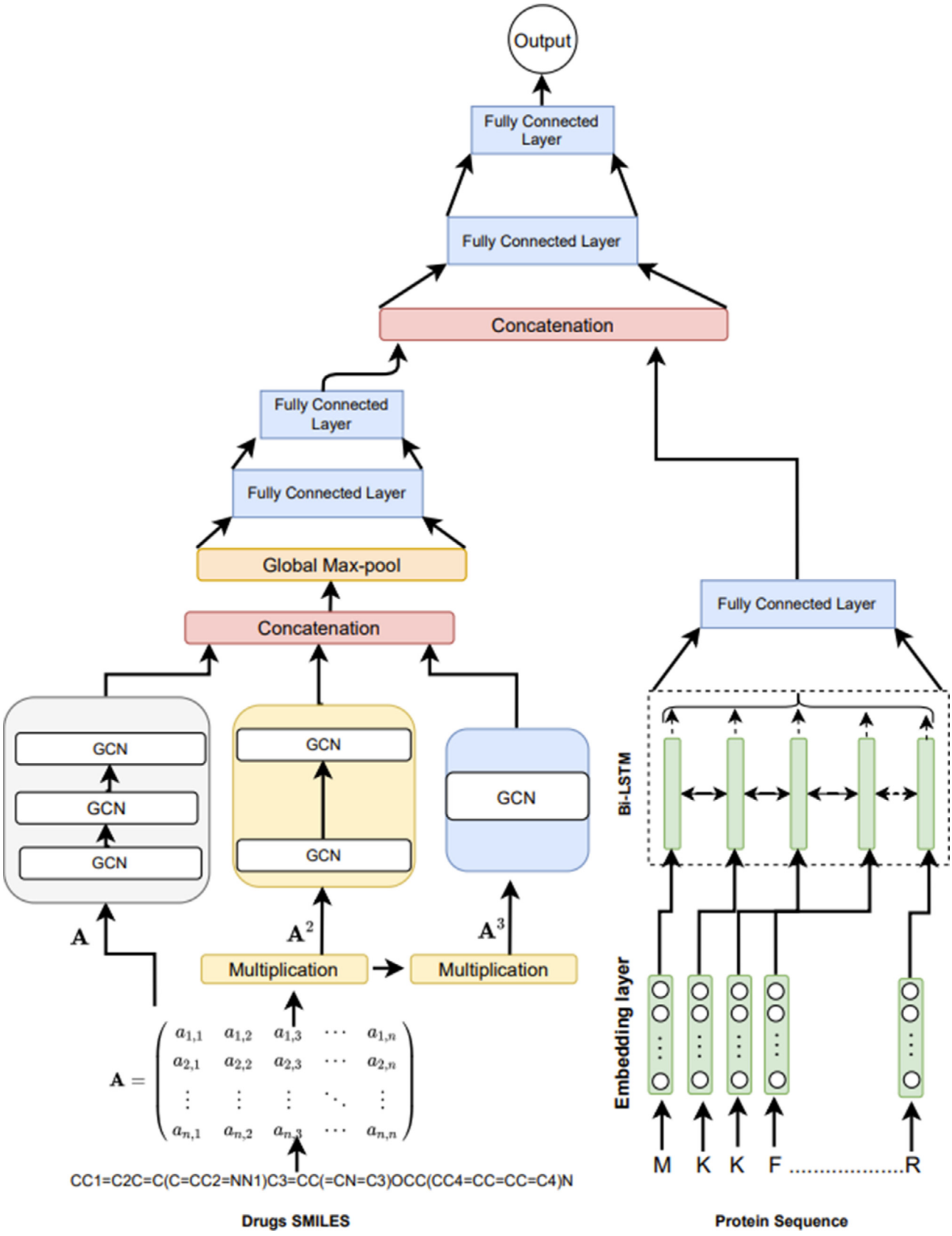

Fig. 11. Model Architecture of DeepGLSTM [126].

Table 3
State of the Art Application of GANs in Drug and Molecular Design.

Study	Model	Approach	Results
Bian et al. [114]	DCGAN	Fingerprints of Molecular Compounds	Screened and developed selective compounds targeting cannabinoid receptors
Zhao et al. [118]	GAN	SMILES	Predicts drug target binding affinity
Guimares et al. [117]	ORGAN	Molecules encoded as SMILES	Ability to Bias generation process towards desired metrics
Gupta et al. [116]	FBGAN	Synthetic DNA Sequence Encodings for Proteins of variable length	Generated proteins with desirable biophysical properties
Rossetto et al. [115]	GANDALF	Peptides for protein targets	Generated peptides comparable to FDA approved drugs and better than Pepcomposer
Anand et al. [113]	DCGAN	ADMM	Quickly producing plausible protein structures
Karimi et al. [112]	gcWGAN	Low-dimensional and generalizable representation of fold space	Comparable or better fold accuracy yet much more sequence diversity and novelty than related models
Sabban et al. [111]	RamaNet	Synthetic Protein Structures	Generated a novel helical backbone topology using an LSTM neural network architecture using only the φ and ψ angles as features
Polykovskiy et al. [127]	ECAAE	Latent Vector	ECAAE developed new molecular compounds that might be used to treat rheumatoid arthritis, psoriasis, and vitiligo.
Prykhodko et al. [128]	LatentGAN	SMILES	LatentGAN was able to produce new drug-like molecules while still being compatible with recurrent neural networks.
Kadurin et al. [129,130]	druGAN	Latent Vector	druGAN developed unique chemical compounds that might be used as anticancer drugs.
Lengeling et al. [131]	ORGANIC	SMILES	ORGANIC performed well in terms of estimating drug-likeness quantitatively.
Putin et al. [132]	RANC	SMILES	In terms of various drug discovery parameters, RANC outperformed ORGANIC.

Table 4
Examples of commonly used molecular representations.

Representation/Fingerprints	Description
SMILES [133]	A specification for describing the structure of chemical species using short ASCII strings.
Canonical SMILES	Provides a method for determining which of all potential SMILES will be utilized
InChI [134]	A textual identifier for chemical substances that provides methods to encode molecular information.
InChI Key	The hashed version of the full InChI.
MACC Keys [135]	166 bit structural key descriptors associated with a SMARTS pattern.
Circular [136,137]	Created by enumerating all circular fragments grown radially from each heavy atom of the molecule up to the given radius.
Path [138]	Created by enumerating linear fragments of a molecular graph up to a given size.
Tree [138]	Generated by enumerating tree fragments of a molecular graph.
Atom Pair [139]	Hashes results from encoding each atom as a type and enumerating distances between atom pairs.

data frameworks. Drug discovery relies on a wide variety of data kinds, many of which are far from exhaustive. One example is that not all protein folds and structures are fully understood, and the data space is only partially covered. Therefore, even if significant progress has been made, applications in which these structures are projected are not yet as good as in other fields. For small molecule synthesis, where the complete chemical space is unknown, this holds true as well for reaction prediction.

Data curation is essential for sharing information that may be used again, yet it can be costly due to the expertise and time involved. There is a fine line between the computational abilities and in-depth biological and domain experience needed for the task of biological curation, which entails extracting biological material from the scientific literature and integrating it into a database.114 Increased access to high-quality data already available in the public domain may be possible through coordinated efforts to create common data resources and metadata (labels). Metadata from unsuccessful as well as successful drug discovery initiatives are included since they can be used to build techniques to predict and identify aspects that can help lower attrition rates. Large data resources of company bioactive data sets of experimental substances and historical clinical trial data require significantly greater pre-competitive collaboration.

Another area where ML models fall short is in making predictions about new paradigms. ML models can only make predictions within the constraints of the training data, as this is the very foundation of the field. For instance, in medicinal chemistry, classical approaches are likely necessary for the design of molecules with different modes of action such as macrocycles, protein–protein interaction inhibitors, or PROTACs.

Machine learning (ML) algorithms, particularly DL techniques, have made AI practical for use in business and daily life. The analysis of omics and imaging data, in particular, has seen the immediate effects of ML approaches' expansion into the drug research and healthcare sectors. Speech recognition, natural language processing, computer vision, and other applications have all seen success with ML algorithms. Among the most frequent examples of such devices are Internet-connected "smart assistants," which may now routinely relay health-related information through the transmission of voice recordings and visuals. Medical decision-making on therapeutic benefits, clinical biomarkers, and adverse effects may be greatly aided by ML techniques applied to data acquired from such a conglomeration of Internet-enabled technologies combined with biological data.

7. Conclusion

Searching chemical space for required functions can be greatly aided by generative machine learning models like GANs. It is difficult to foretell the binding affinity between medication and its target during the drug discovery process. The labeled data upon which supervised systems rely is both costly and challenging to acquire on a large scale, making these approaches impractical for widespread use. We discuss how GANs can be used to train representations from raw sequence data of proteins and medicines, and how convolutional regression can be used to predict the affinity.

Innovations in the many omics fields have ushered in a new era of big data for the drug discovery industry. Such massive amounts of data present exciting prospects for advancing life sciences, but they also raise

Table 5
Well established cheminformatics databases available for drug discovery.

Database	Description	Web Linkage	Usage Example
UniProt [140]	Resource for protein sequence and annotation data.	https://www.uniprot.org	Protein sequence homology search, alignment, and protein ID retrieving especially for structural-based drug discovery.
RCSB PDB [141]	Povides access to 3D structure data for large biological molecules.	https://www.rcsb.org	Protein 3D structures are fundamental for hot spot identification, docking simulation, and molecular dynamics simulation in structural-based drug discovery.
PDBind [142]	Provides a collection of the experimentally measured binding affinity data for all types of biomolecular complexes deposited in the PDB.	http://www.pdbbind.org.cn	The receptor-ligand binding data for resolved protein structures can function as the benchmark to evaluate future simulations.
PubChem [143]	World's largest collection of chemical information.	https://www.pubchem.ncbi.nlm.nih.gov	To acquire comprehensive chemical information ranging from NMR spectra, physical-chemical properties, to biomolecular interactions.
ChEMBL [144]	Curated database containing bioactive molecules with drug-like properties.	https://www.ebi.ac.uk/chembl/	To collect cheminformatics data of reported molecules for a given target. A high-quality compound collection is the key to the ligand-based drug discovery.
SureChEMBL [145]	Collection containing compounds extracted from patent literature.	https://www.surechembl.org/search/	Compound-patent associations
BindingDB [146]	Database of measured binding affinities for the interactions of protein considered to be drug-targets with small, drug-like molecules.	https://www.bindingdb.org/bind/index.jsp	To retrieve compound sets for a specific target similar to ChEMBL but with the focus on experimental binding affinities.
DrugBank [147]	Combines detailed drug data with drug target information.	https://www.drugbank.ca	Drug repurposing study for existing drugs. On-target and off-target analysis for a compound
ZINC [148]	Database of commercially- available compounds.	https://zinc.docking.org	Zinc database is good for virtual screening on hit identification as the compounds are commercially available for quick biological validations afterwards.
Enamine	Provides enumerated database of synthetically feasible molecules.	https://enamine.net	The establishment of a target-specific compound library. Fragment-based drug discovery.
ASD [149]	Resource for structure, function, disease and related annotation for allosteric macromolecules and allosteric modulators.	http://mdl.shsmu.edu.cn/ASD/	To facilitate the research on allosteric modulation with enriched chemical data on allosteric modulators.
GDB [150]	Provides multiple subsets of combinatorially generated compounds.	http://gdb.unibe.ch/downloads/	Using combinatorial chemistry is a good way to largely expand the chemical space.

Table 6
Commonly used cheminformatics and machine learning packages.

Package	Description	Web Linkage
RDKit [151]	RDKit is an open-source toolkit for cheminformatics. Features include 2D and 3D molecular operations, descriptor generation, molecular database cartridge, etc.	https://www.rdkit.org
Open Babel [152]	Open Babel is an open chemical toolbox to search, convert, analyze, or store data from molecular modeling, chemistry, solid- state materials, biochemistry, or related areas.	http://openbabel.org/wiki/Main_Page
CDK [153]	The Chemistry Development Kit (CDK) is a collection of modular Java libraries for processing cheminformatics.	https://www.cdk.github.io
KNIME [154]	KNIME is a workflow environment in data science that can be integrated to automate certain cheminformatics operations.	https://www.knime.com
TensorFlow [155]	TensorFlow is an open-source platform for machine learning. It has a set of tools, libraries, and community resources that enable researchers to build and deploy ML applications.	https://www.tensorflow.org
CNTK [156]	The Cognitive Toolkit (CNTK) is an open-source toolkit for commercial-grade distributed deep learning. It describes neural networks as a series of computational steps via a directed graph.	https://www.github.com/microsoft/CNTK
Theano [157]	Theano is a Python library for defining, optimizing, and evaluating mathematical expressions.	http://deeplearning.net/software/theano/
PyTorch [158]	PyTorch is an open-source machine learning library based on the Torch library.	https://pytorch.org
Keras [159]	Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation	https://www.keras.io
Scikit-Learn[160]	Scikit-learn is a free software machine learning library for the Python programming language.	https://www.scikit-learn.org/stable/

concerns about the veracity and reproducibility of findings. As a means of avoiding potential pitfalls, understanding the current state-of-the-art of research reproducibility in computational drug discovery is crucial. This will guarantee that the underlying work is of high quality and will withstand the reproduction of the described methodology by the external research group. In this overview, we looked at the various methods and tools available for achieving reproducibility in computational drug development initiatives.

Computational drug development is expected to advance thanks to the developing culture of sharing the underlying data and scripts reported in research articles. This will allow for a new and useful knowledge base to be progressively created on top of its predecessors, much

like a snowball. Policies enacted in recent years by funding agencies and publishers favor data and code sharing, which is further facilitated by third-party platforms (e.g. Authorea, Code Ocean, Jupyter notebook, Manuscripts.io, etc.) that further enhance reproducibility by transforming online manuscripts and codes from static files waiting to be downloaded into "living" codes and documents that can be dynamically edited and executed in real-time.

In conclusion, we have tried to describe the wide variety of problems encountered by the predictive modeling community as it performs its mission to create and distribute effective and trustworthy computational tools for the pharmaceutical industry. Evidence presented herein demonstrates the importance of close collaboration among researchers working at the front lines of drug discovery, those responsible for intermediate data modeling, and the administrators and programmers in the back offices. There has to be a better knowledge of these concerns and a shared terminology in order to optimize their impact, yet these groups encounter challenges that are very distinct in nature. Given the scope of the disciplines at play, this is no easy feat. Data modelers, tool developers, and administrators should keep in mind that their tools will be used by front-line scientists in a constantly changing work environment, as we point out. Due to its fluid nature, it may not always align with the recommendations of the data science community (i.e. due to ever-changing needs).

Therefore, it's important to recognize that model developer may disagree with the developer community on the best approach. For instance, while user-derived descriptors (such as experimental data or non-standard 3D computational models) could be ideal, they can be challenging to include swiftly into QSAR models. On the other hand, it is possible that interpretability is more important than raw predictive performance when choosing a predictive model. Because selection requirements are typically driven by statistical considerations rather than the needs of the end user, the latter types of models may not exist in automated solutions in now prevalent modeling workflows.

Transparency in implementations and easy access to validate the analyses are both benefits of open source. It might be challenging to maintain track of the various analysis tools and settings while dealing with data and modeling. As a result, drug discovery workflow solutions are becoming increasingly popular. They aid in producing more trustworthy multi-step computations, as well as in establishing a trail of evidence and making results easy to reproduce. Common Workflow Language is one example of an initiative seeking to standardize and promote interoperability among workflow specification languages.

The usage of public or shared computing infrastructures (HPC/Cloud) is necessitated by the ever-increasing amounts of data, but this introduces a new layer of complexity for computational reproducibility. The usage of virtual machines and software containers has made it possible for all data analysis tools to be used across different platforms. High levels of automation and, by extension, increased repeatability, are possible when workflow systems are integrated with the container and virtual machine infrastructure. For example, when deploying models as networked services, virtual infrastructure and containers allow for more dependable and reproducible service delivery.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

We would like to acknowledge Drexel Society of Artificial Intelligence for its contributions and support for this research.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ailsci.2022.100045.

References

[1] Chen H, Engkvist O, Wang Y, Olivecrona M, Blaschke T. The rise of deep learning in drug discovery. Drug Discov Today 2018;23(6):1241–50.
[2] Vamathevan J, Clark D, Czodrowski P, Dunham I, Ferran E, Lee G, et al. Applications of machine learning in drug discovery and development. Nat Rev Drug Discovery 2019;18(6):463–77.
[3] Hessler G, Baringhaus K-H. Artificial intelligence in drug design. Molecules 2018;23(10):2520.
[4] Butler KT, Davies DW, Cartwright H, Isayev O, Walsh A. Machine learning for molecular and materials science. Nature 2018;559(7715):547–55.
[5] Segler MHS, Preuss M, Waller MP. Planning chemical syntheses with deep neural networks and symbolic AI. Nature 2018;555(7698):604–10.
[6] Baskin II, Winkler D, Tetko IV. A renaissance of neural networks in drug discovery. Expert Opin Drug Discov 2016;11(8):785–95.
[7] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521(7553):436–44.
[8] Hinton G. Deep learning–a technology with the potential to transform health care. JAMA 2018;320(11):1101–2.
[9] Jing Y, Bian Y, Hu Z, Wang L, Xie X-QS. Deep learning for drug design: an artificial intelligence paradigm for drug discovery in the big data era. AAPS J 2018;20(3):1–10.
[10] Rifaioglu AS, Atas H, Martin MJ, Cetin-Atalay R, Atalay V, Doğan T. Recent applications of deep learning and machine intelligence on in silico drug discovery: methods, tools and databases. Brief Bioinformatics 2019;20(5):1878–912.
[11] Google Developers, The discriminator and generator, 2021, https://www.developers.google.com/machine-learning/gan/discriminator
[12] Goh GB, Hodas NO, Vishnu A. Deep learning for computational chemistry. J Comput Chem 2017;38(16):1291–307.
[13] Mamoshina P, Vieira A, Putin E, Zhavoronkov A. Applications of deep learning in biomedicine. Mol Pharm 2016;13(5):1445–54.
[14] Ekins S. The next era: deep learning in pharmaceutical research. Pharm Res 2016;33(11):2594–603.
[15] Kingma D.P., Welling M.. Auto-encoding variational bayes. arXiv preprint arXiv:131261142013;.
[16] Ding J, Condon A, Shah SP. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. Nat Commun 2018;9(1):1–13.
[17] Wang D, Gu J. Vasc: dimension reduction and visualization of single-cell rna-seq data by deep variational autoencoder. Genomics Proteomic Bioinform 2018;16(5):320–31.
[18] Blaschke T, Olivecrona M, Engkvist O, Bajorath J, Chen H. Application of generative autoencoder in de novo molecular design. Mol Inform 2018;37(1–2):1700123.
[19] Ghasemi F, Mehridehnavi A, Perez-Garrido A, Perez-Sanchez H. Neural network and deep-learning algorithms used in QSAR studies: merits and drawbacks. Drug Discov Today 2018;23(10):1784–90.
[20] Ching T, Himmelstein DS, Beaulieu-Jones BK, Kalinin AA, Do BT, Way GP, et al. Opportunities and obstacles for deep learning in biology and medicine. J R Soc Interface 2018;15(141):20170387.
[21] Dana D, Gadhiya SV, St Surin LG, Li D, Naaz F, Ali Q, Paka L, Yamin MA, Narayan M, Goldberg ID, et al. Deep learning in drug discovery and medicine; scratching the surface. Molecules 2018;23(9):2384.
[22] Lin E, Kuo P-H, Liu Y-L, Yu YW-Y, Yang AC, Tsai S-J. A deep learning approach for predicting antidepressant response in major depression using clinical and genetic biomarkers. Front Psychiatry 2018;9:290.
[23] Lin E, Tsai S-J. Machine learning in neural networks. In: Frontiers in psychiatry. Springer; 2019. p. 127–37.
[24] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. Adv Neural Inf Process Syst 2014;27.
[25] Zhao H, Li H, Maurer-Stroh S, Cheng L. Synthesizing retinal and neuronal images with generative adversarial nets. Med Image Anal 2018;49:14–26.
[26] Hu B, Tang Y, Eric I, Chang C, Fan Y, Lai M, et al. Unsupervised learning for cell-level visual representation in histopathology images with generative adversarial networks. IEEE J Biomed Health Inform 2018;23(3):1316–28.
[27] Mardani M, Gong E, Cheng JY, Vasanawala SS, Zaharchuk G, Xing L, et al. Deep generative adversarial neural networks for compressive sensing MRI. IEEE Trans Med Imaging 2018;38(1):167–79.
[28] Lin E, Mukherjee S, Kannan S. A deep adversarial variational autoencoder model for dimensionality reduction in single-cell RNA sequencing analysis. BMC Bioinformatics 2020;21(1):1–11.
[29] Kadurin A, Aliper A, Kazennov A, Mamoshina P, Vanhaelen Q, Khrabrov K, et al. The cornucopia of meaningful leads: applying deep adversarial autoencoders for new molecule development in oncology. Oncotarget 2017;8(7):10883.
[30] Kadurin A, Nikolenko S, Khrabrov K, Aliper A, Zhavoronkov A. DruGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. Mol Pharm 2017;14(9):3098–104.
[31] Alqahtani H, Kavakli-Thorne M, Kumar G. Applications of generative adversarial networks (GANs): an updated review. Arch Comput Methods Eng 2021;28(2):525–52.

[32] Lan L, You L, Zhang Z, Fan Z, Zhao W, Zeng N, et al. Generative adversarial networks and its applications in biomedical informatics. Front Public Health 2020;8:164.
[33] Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville AC. Improved training of wasserstein GANs. Adv Neural Inf Process Syst 2017;30.
[34] Mirza M., Osindero S.. Conditional generative adversarial nets. arXiv preprint arXiv:1411178420142a;.
[35] Makhzani A., Shlens J., Jaitly N., Goodfellow I., Frey B.. Adversarial autoencoders. arXiv preprint arXiv:151105644 2015;.
[36] Rezende DJ, Mohamed S, Wierstra D. Stochastic backpropagation and approximate inference in deep generative models. In: International conference on machine learning. PMLR; 2014. p. 1278–86.
[37] Guimaraes G.L., Sanchez-Lengeling B., Outeiral C., Farias P.L.C., Aspuru-Guzik A.. Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models. arXiv preprint arXiv:170510843
[38] Sanchez-Lengeling B., Outeiral C., Guimaraes G.L., Aspuru-Guzik A.. Optimizing distributions over molecular space. an objective-reinforced generative adversarial network for inverse-design chemistry (ORGANIC)2017a;.
[39] Putin E, Asadulaev A, Ivanenkov Y, Aladinskiy V, Sanchez-Lengeling B, Aspuru-Guzik A, et al. Reinforced adversarial neural computer for de novo molecular design. J Chem Inf Model 2018;58(6):1194–204.
[40] Putin E, Asadulaev A, Vanhaelen Q, Ivanenkov Y, Aladinskaya AV, Aliper A, et al. Adversarial threshold neural computer for molecular de novo design. Mol Pharm 2018;15(10):4386–97.
[41] Polykovskiy D, Zhebrak A, Vetrov D, Ivanenkov Y, Aladinskiy V, Mamoshina P, et al. Entangled conditional adversarial autoencoder for de novo drug discovery. Mol Pharm 2018;15(10):4398–405.
[42] De Cao N., Kipf T.. MolGAN: an implicit generative model for small molecular graphs. arXiv preprint arXiv:180511973 2018;.
[43] Guarino M., Shah A., Rivas P.. DiPol-GAN: generating molecular graphs adversarially with relational differentiable pooling2017;.
[44] Prykhodko O, Johansson SV, Kotsias P-C, Arús-Pous J, Bjerrum EJ, Engkvist O, et al. A de novo molecular generation method using latent vector based generative adversarial network. J Cheminform 2019;11(1):1–13.
[45] Mitchell TM. Machine learning. McGraw-Hill Education; 1997.
[46] Aha D.W. Kibler D. Albert M.K.. Instance-based learning algorithms1991;.
[47] Barlow HB. Unsupervised learning. Neural Comput 1989;1(3):295–311. doi:10.1162/neco.1989.1.3.295. https://www.direct.mit.edu/neco/article-pdf/1/3/295/811863/neco.1989.1.3.295.pdf
[48] Kaelbling LP, Littman ML, Moore AW. Reinforcement learning: a survey. CoRR 1996. cs.AI/9605103, https://www.arxiv.org/abs/cs/9605103
[49] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521(7553):436–44.
[50] Chauvin Y, Rumelhart DE. Backpropagation: theory, architectures, and applications. Psychology Press; 2013.
[51] Geoghegan BD. Orientalism and informatics: alterity from the chess-playing turk to Amazon's mechanical turk. Ex-position 2020(43):45.
[52] Carter F. The turing bombe. Bletchley Park Trust; 2008.
[53] French RM. The turing test: the first 50 years. Trends Cogn Sci 2000;4(3):115–22.
[54] Howard J. Artificial intelligence: implications for the future of work. Am J Ind Med 2019;62(11):917–26.
[55] Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev 1958;65(6):386.
[56] Sharples M, Hogg D, Hutchinson C, Torrance S, Young D. Computers and thought: a practical introduction to artificial intelligence. The MIT Press; 1989.
[57] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE 1998;86(11):2278–324.
[58] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 2012;25.
[59] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. Advances in neural information processing systems, vol 27. Curran Associates, Inc; 2014. https://www.proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf
[60] Chen JX. The evolution of computing: alphago. Comput Sci Eng 2016;18(4):4–7. doi:10.1109/MCSE.2016.74.
[61] Johnson M, Schuster M, Le QV, Krikun M, Wu Y, Chen Z, et al. Google's multilingual neural machine translation system: enabling zero-shot translation. CoRR 2016. http://www.arxiv.org/abs/1611.04558
[62] Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, Tunyasuvunakool K, Bates R, Žídek A, Potapenko A, et al. Highly accurate protein structure prediction with AlphaFold. Nature 2021;596(7873):583–9.
[63] Mak K-K, Pichika MR. Artificial intelligence in drug development: present status and future prospects. Drug Discov Today 2019;24(3):773–80.
[64] Ng A, Jordan M. On discriminative vs. generative classifiers: a comparison of logistic regression and naive bayes. Advances in neural information processing systems, vol 14. MIT Press; 2001. https://proceedings.neurips.cc/paper/2001/file/7b7a53e239400a13bd6be6c91c4f6c4e-Paper.pdf
[65] Mirza M, Osindero S. Conditional generative adversarial nets. CoRR 2014. http://arxiv.org/abs/1411.1784
[66] Denton EL, Chintala S, Szlam A, Fergus R. Deep generative image models using a laplacian pyramid of adversarial networks. CoRR 2015. http://arxiv.org/abs/1506.05751
[67] Radford A., Metz L., Chintala S.. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:151106434 2015;.
[68] Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks. In: International conference on machine learning. PMLR; 2017. p. 214–23.
[69] Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, Abbeel P. InfoGAN: interpretable representation learning by information maximizing generative adversarial nets. CoRR 2016. http://arxiv.org/abs/1606.03657
[70] Zhu J-Y, Park T, Isola P, Efros AA. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision; 2017. p. 2223–32.
[71] Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2019. p. 4401–10.
[72] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. Attention is all you need. CoRR 2017. arXiv preprint arXiv:1706.03762.
[73] Zhang H, Goodfellow I, Metaxas D, Odena A. Self-attention generative adversarial networks. In: International conference on machine learning. PMLR; 2019. p. 7354–63.
[74] Maziarka L, Pocha A, Kaczmarczyk J, Rataj K, Danel T, Warchoł M. Mol-cycleGAN: a generative model for molecular optimization. J Cheminform 2020;12(1):1–18.
[75] Méndez-Lucio O, Baillif B, Clevert D-A, Rouquié D, Wichard J. De novo generation of hit-like molecules from gene expression signatures using artificial intelligence. Nat Commun 2020;11(1):1–10.
[76] Lipton ZC, Berkowitz J, Elkan C. A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:150600019 2015.
[77] Martinez-Mayorga K, Madariaga-Mazon A, Medina-Franco JL, Maggiora G. The impact of chemoinformatics on drug discovery in the pharmaceutical industry. Expert Opin Drug Discov 2020;15(3):293–306.
[78] Weininger D. Smiles, a chemical language and information system. 1. Introduction to methodology and encoding rules. J Chem Inf Comput Sci 1988;28(1):31–6.
[79] Tran L, Yin X, Liu X. Disentangled representation learning GAN for pose-invariant face recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 1415–24.
[80] Yin W., Fu Y., Sigal L., Xue X.. Semi-latent GAN: learning to generate and modify facial images from attributes. arXiv preprint arXiv:170402166 2017.
[81] Larsen ABL, Sønderby SK, Larochelle H, Winther O. Autoencoding beyond pixels using a learned similarity metric. In: International conference on machine learning. PMLR; 2016. p. 1558–66.
[82] Zhang Z, Song Y, Qi H. Age progression/regression by conditional adversarial autoencoder. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 5810–18.
[83] Wu H, Zheng S, Zhang J, Huang K. GP-GAN: towards realistic high-resolution image blending. In: Proceedings of the 27th ACM international conference on multimedia; 2019. p. 2487–95.
[84] Zhu J-Y, Park T, Isola P, Efros AA. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision; 2017. p. 2223–32.
[85] Kim T, Cha M, Kim H, Lee JK, Kim J. Learning to discover cross-domain relations with generative adversarial networks. In: International conference on machine learning. PMLR; 2017. p. 1857–65.
[86] Wang Y, Wu L. Beyond low-rank representations: orthogonal clustering basis reconstruction with optimized graph structure for multi-view spectral clustering. Neural Netw 2018;103:1–8.
[87] Liu M-Y, Tuzel O. Coupled generative adversarial networks. Adv Neural Inf Process Syst 2016;29.
[88] Yang D, Xiong T, Xu D, Huang Q, Liu D, Zhou SK, Xu Z, Park J, Chen M, Tran TD, et al. Automatic vertebra labeling in large-scale 3D CT using deep image-to-image network with message passing and sparsity regularization. In: International conference on information processing in medical imaging. Springer; 2017. p. 633–44.
[89] Xue Y, Xu T, Zhang H, Long LR, Huang X. SegAN: adversarial network with multi-scale l1 loss for medical image segmentation. Neuroinformatics 2018;16(3):383–92.
[90] Mogren O.. C-RNN-GAN: continuous recurrent neural networks with adversarial training. arXiv preprint arXiv:161109904 2016;.
[91] Yu H-X, Wu A, Zheng W-S. Cross-view asymmetric metric learning for unsupervised person re-identification. In: Proceedings of the IEEE international conference on computer vision; 2017. p. 994–1002.
[92] Li D, Chen X, Zhang Z, Huang K. Learning deep context-aware features over body and latent parts for person re-identification. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 384–93.
[93] Li J, Liang X, Wei Y, Xu T, Feng J, Yan S. Perceptual generative adversarial networks for small object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 1222–30.
[94] Bjerrum EJ, Sattarov B. Improving chemical autoencoder latent space and molecular de novo generation diversity with heteroencoders. Biomolecules 2018;8(4):131.
[95] Zhou S., Xiao T., Yang Y., Feng D., He Q., He W.. GeneGAN: learning object transfiguration and attribute subspace from unpaired data. arXiv preprint arXiv:170504932 2017.
[96] Qian X, Fu Y, Xiang T, Wang W, Qiu J, Wu Y, Jiang Y-G, Xue X. Pose-normalized image generation for person re-identification. In: Proceedings of the European conference on computer vision (ECCV); 2018. p. 650–67.
[97] Liu J., Li W., Pei H., Wang Y., Qu F., Qu Y., Chen Y.. Identity preserving generative adversarial network for cross-domain person re-identification. IEEE Access2019; 7:114021–114032.
[98] Ledig C, Theis L, Huszár F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z, et al. Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 4681–90.

[99] Tulyakov S, Liu M-Y, Yang X, Kautz J. MoCoGAN: decomposing motion and content for video generation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 1526–35.
[100] Walker J, Marino K, Gupta A, Hebert M. The pose knows: video forecasting by generating pose futures. In: Proceedings of the IEEE international conference on computer vision; 2017. p. 3332–41.
[101] Vondrick C, Pirsiavash H, Torralba A. Generating videos with scene dynamics. Adv Neural Inf Process Syst 2016;29.
[102] Zhou S.-F., Zhong W.-Z.. Drug design and discovery: principles and applications. 2017.
[103] Goh G.B., Siegel C., Vishnu A., Hodas N.O., Baker N.. Chemception: a deep neural network with minimal chemistry knowledge matches the performance of expert-developed QSAR/QSPR models. arXiv preprint arXiv:170606668920172b;.
[104] You J, Liu B, Ying Z, Pande V, Leskovec J. Graph convolutional policy network for goal-directed molecular graph generation. Adv Neural Inf Process Syst 2018;31.
[105] Durant JL, Leland BA, Henry DR, Nourse JG. Reoptimization of MDL keys for use in drug discovery. J Chem Inf Comput Sci 2002;42(6):1273–80.
[106] Sterling T, Irwin JJ. Zinc 15–ligand discovery for everyone. J Chem Inf Model 2015;55(11):2324–37.
[107] Blanchard AE, Stanley C, Bhowmik D. Using GANs with adaptive training data to search for new molecules. J Cheminform 2021;13(1):1–8.
[108] Simonovsky M, Komodakis N. GraphVAE: towards generation of small graphs using variational autoencoders. In: International conference on artificial neural networks. Springer; 2018. p. 412–22.
[109] Lin E, Lin C-H, Lane H-Y. Relevant applications of generative adversarial networks in drug design and discovery: molecular de novo design, dimensionality reduction, and de novo peptide and protein design. Molecules 2020;25(14):3250.
[110] Mouchlis VD, Afantitis A, Serra A, Fratello M, Papadiamantis AG, Aidinis V, et al. Advances in de novo drug design: from conventional to machine learning methods. Int J Mol Sci 2021;22(4):1676.
[111] Sabban S., Markovsky M.. RamaNet: computational de novo helical protein backbone design using a long short-term memory generative adversarial neural network [version 1; peer review: 1 not]2020;.
[112] Karimi M, Zhu S, Cao Y, Shen Y. De novo protein design for novel folds using guided conditional wasserstein generative adversarial networks (gcWGAN). bioRxiv 2019:769919.
[113] Anand N, Huang P. Generative modeling for protein structures. Adv Neural Inf Process Syst 2018;31.
[114] Bian Y, Wang J, Jun JJ, Xie X-Q. Deep convolutional generative adversarial network (dcGAN) models for screening and design of small molecules targeting cannabinoid receptors. Mol Pharm 2019;16(11):4451–60.
[115] Rossetto AM, Zhou W. GANDALF: a prototype of a GAN-based peptide design method. In: Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics; 2019. p. 61–6.
[116] Gupta A., Zou J.. Feedback GAN (FBGAN) for DNA: a novel feedback-loop architecture for optimizing protein functions. arXiv preprint arXiv:180401694 2018;.
[117] Guimaraes G.L., Sanchez-Lengeling B., Outeiral C., Farias P.L.C., Aspuru-Guzik A.. Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models. arXiv preprint arXiv:170510843
[118] Zhao L, Wang J, Pang L, Liu Y, Zhang J. GANsDTA: predicting drug-target binding affinity using GANs. Front Genet 2020:1243.
[119] Nantasenamat C.. Conceptual map of computational drug discovery [CC-BY]. 2019.
[120] Chen H, Engkvist O, Wang Y, Olivecrona M, Blaschke T. The rise of deep learning in drug discovery. Drug Discov Today 2018;23(6):1241–50.
[121] Zhavoronkov A.. Artificial intelligence for drug discovery, biomarker development, and generation of novel chemistry. 2018.
[122] Sanchez-Lengeling B, Aspuru-Guzik A. Inverse molecular design using machine learning: generative models for matter engineering. Science 2018;361(6400):360–5.
[123] Kuhn B, Guba W, Hert J, Banner D, Bissantz C, Ceccarelli S, et al. A real-world perspective on molecular design: miniperspective. J Med Chem 2016;59(9):4087–102.
[124] Vanhaelen Q, Lin Y-C, Zhavoronkov A. The advent of generative chemistry. ACS Med Chem Lett 2020;11(8):1496–505.
[125] Torchilin VP. Drug targeting. Eur J Pharm Sci 2000;11:S81–91.
[126] Mukherjee S., Ghosh M., Basuchowdhuri P.. Deep graph convolutional network and LSTM based approach for predicting drug-target binding affinity. arXiv preprint arXiv:220106872 2022;.
[127] Polykovskiy D, Zhebrak A, Vetrov D, Ivanenkov Y, Aladinskiy V, Mamoshina P, et al. Entangled conditional adversarial autoencoder for de novo drug discovery. Mol Pharm 2018;15(10):4398–405.
[128] Prykhodko O, Johansson SV, Kotsias P-C, Arús-Pous J, Bjerrum EJ, Engkvist O, et al. A de novo molecular generation method using latent vector based generative adversarial network. J Cheminform 2019;11(1):1–13.
[129] Kadurin A, Aliper A, Kazennov A, Mamoshina P, Vanhaelen Q, Khrabrov K, et al. The cornucopia of meaningful leads: applying deep adversarial autoencoders for new molecule development in oncology. Oncotarget 2017;8(7):10883.
[130] Kadurin A, Nikolenko S, Khrabrov K, Aliper A, Zhavoronkov A. DruGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. Mol Pharm 2017;14(9):3098–104.
[131] Sanchez-Lengeling B., Outeiral C., Guimaraes G.L., Aspuru-Guzik A.. Optimizing distributions over molecular space. an objective-reinforced generative adversarial network for inverse-design chemistry (ORGANIC)2017b;.
[132] Putin E, Asadulaev A, Ivanenkov Y, Aladinskiy V, Sanchez-Lengeling B, Aspuru-Guzik A, et al. Reinforced adversarial neural computer for de novo molecular design. J Chem Inf Model 2018;58(6):1194–204.
[133] Weininger D. Smiles, a chemical language and information system. 1. Introduction to methodology and encoding rules. J Chem Inf Comput Sci 1988;28(1):31–6.
[134] Heller SR, McNaught A, Pletnev I, Stein S, Tchekhovskoi D. InChi, the IUPAC international chemical identifier. J Cheminform 2015;7(1):1–34.
[135] Durant JL, Leland BA, Henry DR, Nourse JG. Reoptimization of MDL keys for use in drug discovery. J Chem Inf Comput Sci 2002;42(6):1273–80.
[136] Rogers D, Hahn M. Extended-connectivity fingerprints. J Chem Inf Model 2010;50(5):742–54.
[137] Glen RC, Bender A, Arnby CH, Carlsson L, Boyer S, Smith J. Circular fingerprints: flexible molecular descriptors with applications from physical chemistry to ADME. IDrugs 2006;9(3):199.
[138] Hert J, Willett P, Wilton DJ, Acklin P, Azzaoui K, Jacoby E, et al. Comparison of fingerprint-based methods for virtual screening using multiple bioactive reference structures. J Chem Inf Comput Sci 2004;44(3):1177–85.
[139] Pérez-Nueno VI, Rabal O, Borrell JI, Teixidó J. APIF: a new interaction fingerprint based on atom pairs and its application to virtual screening. J Chem Inf Model 2009;49(5):1245–60.
[140] Apweiler R, Bairoch A, Wu CH, Barker WC, Boeckmann B, Ferro S, et al. UniProt: the universal protein knowledgebase. Nucleic Acids Res 2004;32(suppl_1):D115–19.
[141] Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, et al. The protein data bank. Nucleic Acids Res 2000;28(1):235–42.
[142] Wang R, Fang X, Lu Y, Yang C-Y, Wang S. The PDBbind database: methodologies and updates. J Med Chem 2005;48(12):4111–19.
[143] Kim S, Thiessen PA, Bolton EE, Chen J, Fu G, Gindulyte A, et al. PubChem substance and compound databases. Nucleic Acids Res 2016;44(D1):D1202–13.
[144] Gaulton A, Hersey A, Nowotka M, Bento AP, Chambers J, Mendez D, et al. The chEMBL database in 2017. Nucleic Acids Res 2017;45(D1):D945–54.
[145] Papadatos G, Davies M, Dedman N, Chambers J, Gaulton A, Siddle J, et al. SurechEMBL: a large-scale, chemically annotated patent document database. Nucleic Acids Res 2016;44(D1):D1220–8.
[146] Gilson MK, Liu T, Baitaluk M, Nicola G, Hwang L, Chong J. BindingDB in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology. Nucleic Acids Res 2016;44(D1):D1045–53.
[147] Wishart DS, Feunang YD, Guo AC, Lo EJ, Marcu A, Grant JR, et al. Drugbank 5.0: a major update to the drugbank database for 2018. Nucleic Acids Res 2018;46(D1):D1074–82.
[148] Irwin JJ, Shoichet BK. Zinc- a free database of commercially available compounds for virtual screening. J Chem Inf Model 2005;45(1):177–82.
[149] Huang Z, Mou L, Shen Q, Lu S, Li C, Liu X, et al. ASD v2. 0: updated content and novel features focusing on allosteric regulation. Nucleic Acids Res 2014;42(D1):D510–16.
[150] Ruddigkeit L, Van Deursen R, Blum LC, Reymond J-L. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. J Chem Inf Model 2012;52(11):2864–75.
[151] Landrum G.. Rdkit: Open-source cheminformatics software2016;https://www.github.com/rdkit/rdkit/releases/tag/Release_2016_09_4.
[152] O'Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR. Open babel: an open chemical toolbox. J Cheminform 2011;3(1):1–14.
[153] Willighagen EL, Mayfield JW, Alvarsson J, Berg A, Carlsson L, Jeliazkova N, et al. The chemistry development kit (CDK) v2. 0: atom typing, depiction, molecular formulas, and substructure searching. J Cheminform 2017;9(1):1–19.
[154] Arabie P., Baier N.D., Critchley C.F., Keynes M.. Studies in classification, data analysis, and knowledge organization2006;.
[155] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, et al. {TensorFlow}: a system for {Large-Scale} machine learning. In: 12th USENIX symposium on operating systems design and implementation (OSDI 16); 2016. p. 265–83.
[156] Etaati L. Deep learning tools with cognitive toolkit (CNTK). In: Machine learning with microsoft technologies. Springer; 2019. p. 287–302.
[157] Al-Rfou R, Alain G, Almahairi A, Angermueller C, Bahdanau D, Ballas N, et al. Theano: a python framework for fast computation of mathematical expressions. arXiv e-prints 2016:1605.
[158] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. Pytorch: an imperative style, high-performance deep learning library. Adv Neural Inf Process Syst 2019;32.
[159] Ketkar N. Introduction to keras. In: Deep learning with Python. Springer; 2017. p. 97–111.
[160] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: machine learning in python. J Mach Learn Res 2011;12:2825–30.
[161] Norgeot B, Glicksberg BS, Butte AJ. A call for deep-learning healthcare. Nat Med 2019;25(1):14–15.
[162] Esteva A, Robicquet A, Ramsundar B, Kuleshov V, DePristo M, Chou K, et al. A guide to deep learning in healthcare. Nat Med 2019;25(1):24–9.
[163] Yang Z, Huang Y, Jiang Y, Sun Y, Zhang Y-J, Luo P. Clinical assistant diagnosis for electronic medical record based on convolutional neural network. Sci Rep 2018;8(1):1–9.
[164] Mohr DC, Zhang M, Schueller SM. Personal sensing: understanding mental health using ubiquitous sensors and machine learning. Annu Rev Clin Psychol 2017;13:23.
[165] Gkotsis G, Oellrich A, Velupillai S, Liakata M, Hubbard TJP, Dobson RJB, et al. Characterisation of mental health conditions in social media using informed deep learning. Sci Rep 2017;7(1):1–11.
[166] Koscielny S. Why most gene expression signatures of tumors have not been useful in the clinic. Sci Transl Med 2010;2(14). 14ps2–14ps2