IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 October 2023, accepted 14 December 2023, date of publication 22 December 2023, date of current version 7 February 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3346273

SURVEY

A Review of Generative Adversarial Networks (GANs) and Its Applications in a Wide Variety of Disciplines: From Medical to Remote Sensing

ANKAN DASH, JUNYI YE, AND GUILING WANG
New Jersey Institute of Technology, Newark, NJ 07102, USA

Corresponding author: Ankan Dash (ad892@njit.edu)

This work was supported in part by the Federal Highway Administration Exploratory Advanced Research under Grant 693JJ320C000021.

ABSTRACT We look into Generative Adversarial Network (GAN), its prevalent variants and applications in a number of sectors. GANs combine two neural networks that compete against one another using zero-sum game theory, allowing them to create much crisper and discrete outputs. GANs can be used to perform image processing, video generation and prediction, among other computer vision applications. GANs can also be utilised for a variety of science-related activities, including protein engineering, astronomical data processing, remote sensing image dehazing, and crystal structure synthesis. Other notable fields where GANs have made gains include finance, marketing, fashion design, sports, and music. Therefore in this article we provide a comprehensive overview of the applications of GANs in a wide variety of disciplines. We first cover the theory supporting GAN, GAN variants, and the metrics to evaluate GANs. Then we present how GAN and its variants can be applied in twelve domains, ranging from STEM fields, such as astronomy and biology, to business fields, such as marketing and finance, and to arts, such as music. As a result, researchers from other fields may grasp how GANs work and apply them to their own study. To the best of our knowledge, this article provides the most comprehensive survey of GAN's applications in different field.

INDEX TERMS Deep learning, generative adversarial networks, computer vision, time series, applications.

I. INTRODUCTION

Generative Adversarial Networks [1] or GANs belong to the family of Generative models [2]. Generative Models try to learn a probability density function from a training set and then generate new samples that are drawn from the same distribution. GANs generate new synthetic data that resembles real data by pitting two neural networks (the Generator and the Discriminator) against each other. The Generator tries to capture the true data distribution for generating new samples. The Discriminator, on the other hand, is usually a Binary classifier that tries to discern between actual and fake generated samples as precisely as possible.

Over the last few years, GANs have made substantial progress. Due to hardware advances, we can now train deeper and more sophisticated Generator and Discriminator neural network architectures with increased model capacity. GANs have a number of distinct advantages over other types of generative models. Unlike Boltzmann machines [3], GANs do not require Monte Carlo approximations in order to train, and GANs use back-propagation and do not require Markov chains.

GANs have gained a lot of traction in recent years and have been widely employed in a variety of disciplines, with the list of fields in which GANs can be used fast expanding. GANs can be used for data generation and augmentation([4], [5]), image to image translation([6], [7]), image super resolution ([8], [9]) to name a few. Figure 1. shows some of the most widely used GAN applications. It is this versatile nature, that

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Olague.

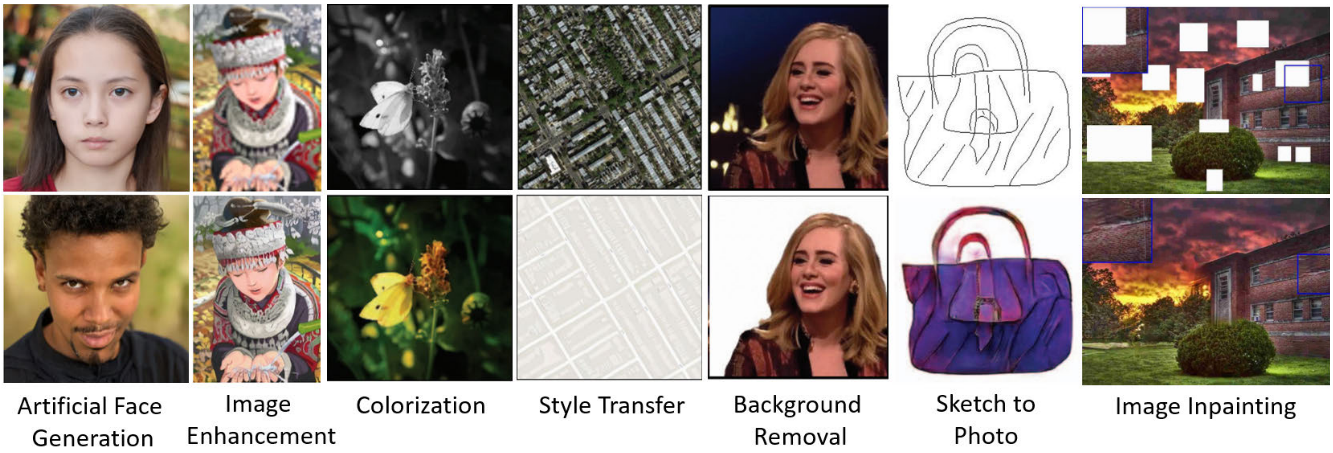


FIGURE 1. Common Uses of GANs in various applications. From left to right: Face generation [10], Image enhancement or super resolution [11], Image Colorization [6], Style transfer [6], sketch to photo [6], image inpainting [12].

has allowed GANs to be applied in completely non-aligned domains such as medicine and astronomy.

There have been a few surveys and reviews about GANs due to their tremendous popularity and importance. However, the majority of past papers have concentrated on two distinct aspects: first, describing GANs and their growth over time, and second, discussing GANs' use in image processing and computer vision applications([13], [14], [15], [16], [17]). As a consequence, the focus has been less on describing GAN applications in a wide range of disciplines. Therefore, we'll present a comprehensive review of GANs in this first-of-its-kind article. We'll look at GANs and some of the most widely used GAN models and variants, as well as a number of evaluation metrics, GAN applications in a variety of 12 areas (including image and video related tasks, medical and healthcare, biology, astronomy, remote sensing, material science, finance, marketing, fashion, sports and music), GAN challenges and limitations, and GAN future directions. Figure 2 shows the overall content presented in the paper.

Some of the major contributions of the paper are highlighted below:

- Describe the wide range of GAN applications in engineering, science, social science, business, art, music and sports. As far as we know, this is the first review paper to cover GAN applications in such diverse domains. This review will assist researchers of various backgrounds in comprehending the operation of GANs and discovering about their wide array of applications.
- Evaluation of GANs include both qualitative and quantitative methods. This survey provides a comprehensive presentation of quantitative metrics that are used to evaluate the performance of GANs in both computer vision and time series data analysis. We include evaluation metrics for GANs' application in time series data which are not discussed in other GAN survey paper. To the best of our knowledge, this is the first survey paper to present time series data evaluation metrics for GANs.

We have organized the rest of the article as follows: Section II presents the basic working of GANs, and the most commonly used GAN variants and their descriptions. Section III summarises some of the frequently used GAN evaluation metrics. Section IV describes the extensive range of applications of GANs in a wide variety of domains. We also provide a table at the end of each subsection summarizing the application area and the corresponding GAN models used. Section V discusses some of the difficulties and challenges that are encountered during the training of GANs. Apart from this we present a short summary concerning the future direction of GAN development. Section VI provides concluding remarks.

II. GAN, GAN VARIANTS AND EXTENSIONS

In this section we describe about GANs, the most common GAN models and extensions. Following a description of GAN theory, we go over twelve GAN variants that serve as foundations or building blocks for many other GAN models. There are a lot of articles on GANs, and a lot of them have named-GANs, which are models that have a specific name that usually contains the word "GAN". We've focused on twelve specific GAN variants. The reader will obtain a better knowledge of the core aspects of GANs by reading through these twelve GAN variants, which will help them navigate other GAN models.

A. GAN BASICS

Generative Adversarial Networks were developed by Ian Goodfellow et al. [1] in the year 2014. GANs belong to the class of Generative models [2]. GANs are based on the min-max, zero-sum game theory. For this, GANs consists of two neural networks: one is the Generator and the other is the Discriminator. The goal of the Generator is to learn to generate fake sample distribution to deceive

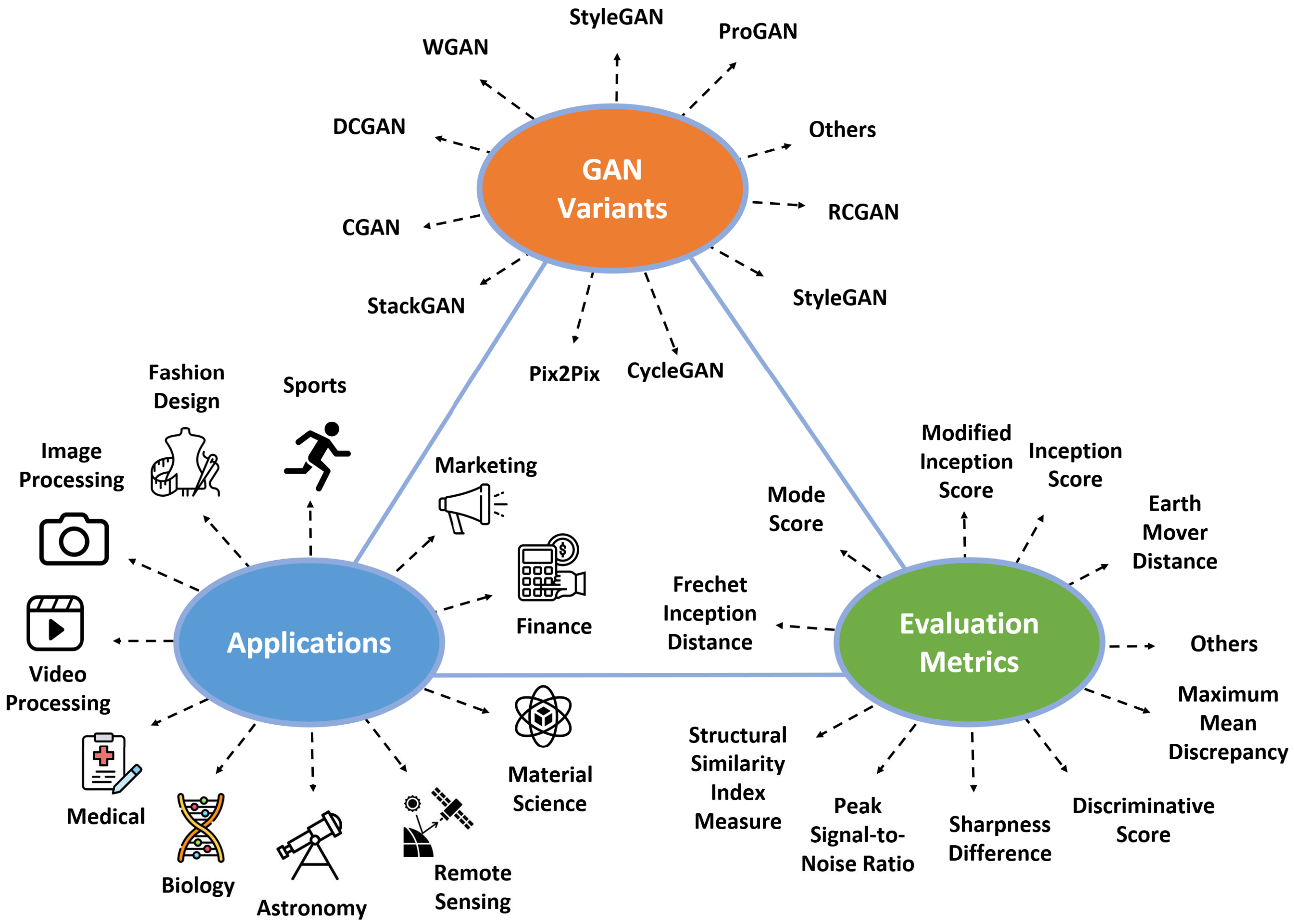


FIGURE 2. GAN variants, application domains and evaluation metrics presented in the paper.

the Discriminator whereas the goal of the Discriminator is to learn to distinguish between real and fake distribution generated by the Generator.

1) NETWORK ARCHITECTURE AND LEARNING

The general architecture of GAN which is comprised of the Generator and the Discriminator is shown in Figure 3. The Generator (G) takes in as input some random noise vector Z and then tries to generate an image using this noise vector indicated as G(z). The generated image is then passed to the Discriminator and based on the output of the Discriminator the parameters of the Generator are updated. The Discriminator (D) is a binary classifier which simultaneously takes a look at both real and fake samples generated by the Generator and ties to decide which ones are real and which ones are fake. That is for a sample image X the Discriminator models the probability of the image being fake or real. The probabilities are then passed back to the Generator as feedback.

Over time each of the Generator and the Discriminator model tries to one up each other by competing against each other this where the term "adversarial" of Generative Adversarial Networks comes from and the optimization is based on the minimax game problem. During training both the Generator's and Discriminator's parameters are updated using back propagation with the ultimate goal of the Generator is to be able to generate realistic looking images and the Discriminator to get progressively better at detecting generated fake images from real ones.

GANs use the Minimax loss function which was introduced by Goodfellow et al. when they introduced GANs for the first time. The Generator tries to minimize the following function while the Discriminator tries to maximize it. The Minimax loss is given as,

$$Min_G \, Max_D f(D, G) = \mathbb{E}_x[log(D(x))] + \mathbb{E}_z[log(1 - D(G(z)))]. \quad (1)$$

Here, E_x is the expected value over all real data samples, $D(x)$ is the probability estimate of the Discriminator if x is real, $G(z)$ is the output of the Generator for a given random noise vector z as input, $D(G(z))$ is the Discriminator's probability estimate if the fake generated sample is real, E_z is the expected value over all random inputs to the Generator.

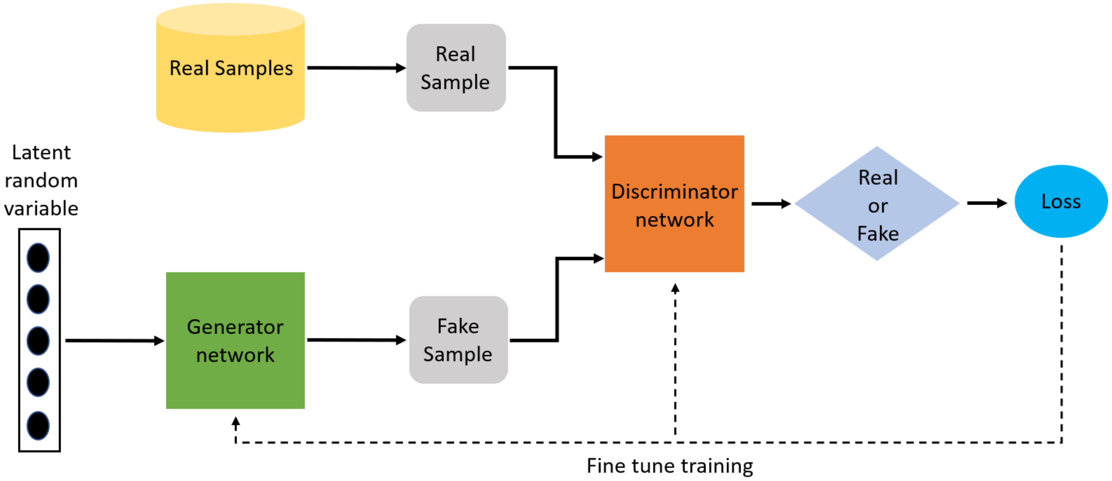


FIGURE 3. Basic GAN architecture.

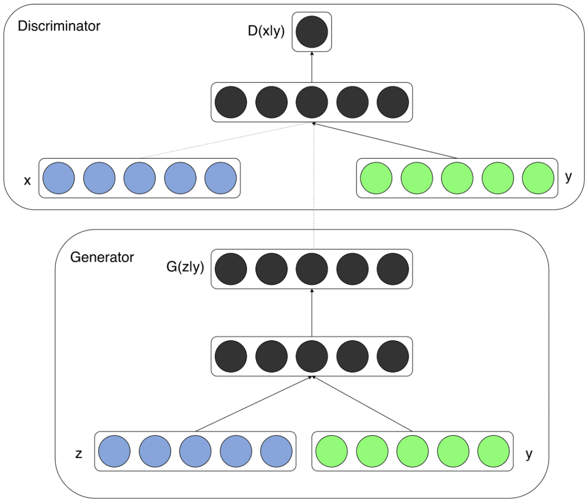


FIGURE 4. cGAN architecture [18].

B. CONDITIONAL GENERATIVE ADVERSARIAL NETS (cGAN)

Conditional Generative Adversarial Nets [18] or cGANs are an extension of GANs for conditional sample generation. This gives control over the modes of data being generated. cGANs use some extra information y, such as class labels or other modalities, to perform conditioning by concatenating this extra information y with the input and feeding it into both the Generator G and the Discriminator D. The Minimax objective function can be modified as shown below,

$$\min_G \max_D f(D, G) \\ = \mathbb{E}_x[\log(D(x|y))] + \mathbb{E}_z[\log(1 - D(G(z|y)))] \quad (2)$$

C. WASSERSTEIN GAN (WGAN)

The authors of WGAN [19] introduced a new algorithm which gave an alternative to traditional GAN training. They showed that their new algorithm improved the stability of model learning and prevent problems such as mode collapse. For the critique model, WGAN uses weight clipping, which ensures that weight values (model parameters) stay within pre-defined ranges. The authors found that Jensen-Shannon divergence is not ideal for measuring the distance of the distribution of the disjoint parts. Therefore they used the Wasserstein distance which uses the concept of Earth mover's(EM) distance instead to measure the distance between the generated and the real data distribution and while training the model tries to maintain One-Lipschitz continuity [20].

The EM or Wasserstein distance for the real data distribution P_r and the generated data distribution P_g is given as

$$W(P_r, P_g) = inf_{\gamma \varepsilon \Pi(P_r, P_g)} \mathbb{E}_{(x,y)\sim r}[\|x - y\|] \quad (3)$$

where $\Pi(P_r, P_g)$ denotes the set of all joint distributions $\gamma(x, y)$ whose marginals are respectively P_r and P_g. However, the equation for the Wasserstein distance is highly intractable. Therefore the authors used the Kantorovich-Rubinstein duality to approximate the Wasserstein distance as

$$max_{w\varepsilon\omega} \mathbb{E}_{x\sim P_r}[f_w(x)] - \mathbb{E}_{z\sim p(z)}[f_w(G(z))] \quad (4)$$

where $(f_w)_{w\varepsilon\omega}$ represents a parameterized family of functions that are all K-Lipschitz for some K. The Discriminator D's goal is to optimize this parameterized function which represents the approximated Wasserstein distance. The goal of the Generator G is to miminize the above Wasserstein distance equation such that the generated data distribution is as close as possible to the real distribution. The overall WGAN objective function is given as

$$min_G max_{w\varepsilon\omega} \mathbb{E}_{x\sim P_r}[f_w(x)] - \mathbb{E}_{z\sim p(z)}[f_w(G(z))] \quad (5)$$

or

$$min_G max_D \mathbb{E}_{x\sim P_r}[f_w(x)] - \mathbb{E}_{z\sim p(z)}[f_w(G(z))] \quad (6)$$

Even though WGAN improved training stability and alleviated problems such as mode collapse however enforcing the

Lipschitz constraint is a challenging task. WGAN-GP [20] proposes an alternative to clipping weights by using gradient penalty to penalize the norm of gradient of the critic with respect to its input.

D. UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS (DCGANs)

Radford et al. [5] introduced the deep convolutional generative adversarial networks or DCGANs. As the name suggests DCGANs use deep convolutional neural networks for both the Generator and Discriminator models. The original GAN architecture used only multi-layer perceptrons or MLPs but since CNNs are better at images than MLP, the authors of DCGAN used CNN in the Generator G and Discriminator D neural network architecture. Thre key features of the DCGANs neural network architecture are: (a) First, for the Generator, convolutions are replaced with transposed convolutions, so the representation at each layer of the Generator is successively larger, as it maps from a low-dimensional latent vector onto a high-dimensional image. Replacing any pooling layers with strided convolutions (Discriminator) and fractional-strided convolutions (Generator). (b) Second, use batch normalization in both the Generator and the Discriminator. (c) Third, use ReLU activation in Generator for all layers except for the output, which uses Tanh. Use LeakyReLU activation in the Discriminator for all layers. (d) Fourth, use the Adam optimizer instead of SGD with momentum. All of these modifications rendered DCGAN to achieve stable training. DCGAN was important because the authors demonstrated that by enforcing certain constraints we can develop complex high quality Generators. The authors also made several other modifications to the vanilla GAN architecture.

E. PROGRESSIVE GROWING OF GANs FOR IMPROVED QUALITY, STABILITY, AND VARIATION (ProGAN)

Karras et al. [4] introduced a new training methodology for training GANs to generate high resolution images. The idea behind ProGAN is to be able to synthesize high resolution and high quality images via the incremental (gradual) growing of the Discriminator and the Generator networks during the training process. ProGAN makes it easier for the Generator to generate higher resolution images by gradually training it from lower resolution images to those higher resolution images. That is in a progressive GAN, the Generator's first layers produce very low-resolution images, and subsequent layers add details. Training is considerably stabilised by the progressive learning process.

F. INTERPRETABLE REPRESENTATION LEARNING BY INFORMATION MAXIMIZING GENERATIVE ADVERSARIAL NETS (InfoGAN)

The key motivation behind InfoGAN [21] is to enable GANs to learn disentangled representations and have control over the properties or features of the generated images in an unsupervised manner. To do this instead of using just a noize vector z as input the authors decompose the noise vector into two parts first being the traditional noise vector z and second as new "latent code vector" c. This code has a predictable effect on the output images. The objective function for InfoGAN [21] is given as,

$$Min_G Max_D f_1(D, G) = f(D, G) - \lambda I(c; G(z, c)) \tag{7}$$

where λ is the regularization parameter, $I(c; G(z, c))$ is the mutual information between the latent code c and the Generator output $G(z, c)$. The idea is to maximize the mutual information between the latent code and the Generator output. This encourages the latent code c to contain as much as possible, important and relevant features of the real data distributions. However it is not practical to calculate the mutual information $I(c; G(z, c))$ explicitly as it requires the posterior $P(c|x)$, therefore a lower bound for $I(c; G(z, c))$ is approximated. This can be achieved by defining an auxiliary distribution $Q(c|x)$ to approximate $P(c|x)$. Thus the final form of the objective function is then given by this lower-bound approximation to the Mutual Information:

$$Min_G Max_D f_1(D, G) = f(D, G) - \lambda L_1(c; Q) \tag{8}$$

where $L_1(c; Q)$ is the lower bound for $I(c; G(z, c))$. If we compare the above equation to the original GAN objective function we realize that this framework is implemented by merely adding a regularization term to the original GAN's objective function.

G. StackGAN: TEXT TO PHOTO-REALISTIC IMAGE SYNTHESIS WITH STACKED GENERATIVE ADVERSARIAL NETWORKS (StackGAN)

StackGAN [22], takes in as input a text description and then synthesizes high quality images using the given text description. The authors proposed StackGAN to generate 256×256 photo-realistic images based on text descriptions. To generate photo-realistic images StackGAN uses a sketch-refinement process, StackGAN decomposes the difficult problem into more manageable sub-problems by using Stacked Generative Adversarial Networks. The **Stage-I GAN** creates Stage-I low-resolution images by sketching the object's primitive shape and colours based on the given text description. The **Stage-II GAN** generates high-resolution images with photo-realistic details using Stage-I results and text descriptions as inputs.

To be able to do this, StackGAN architecture consists of the following components: (a) Input variable length text description is converted into a fixed length vector embedding. (b) Conditioning Augmentation. (c) Stage I Generator: Generates (128 × 128) images (d) Stage I Discriminator (e) Stage II Generator: Generates (256 × 256) images. (f) Stage II Discriminator. The variable length text description is first converted to a vector embedding which is non-linearly transformed to generate conditioning latent variables as the input of the Generator. Filling the latent space of the embedding with randomly generated fillers

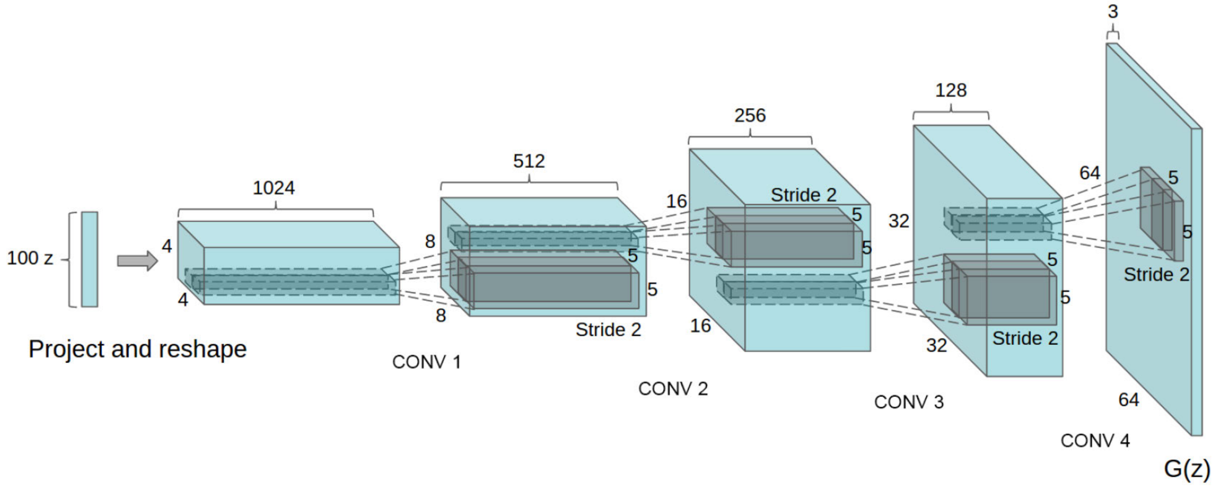


FIGURE 5. **DCGAN generator architecture [5].**

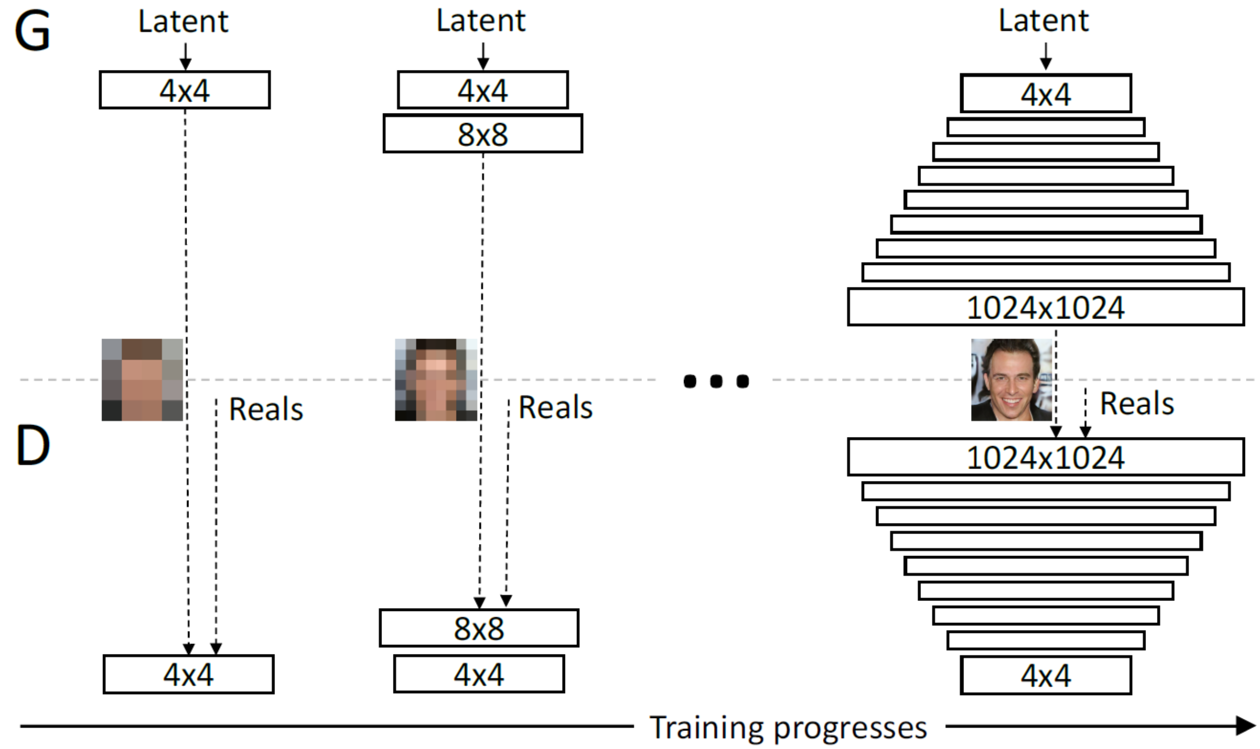


FIGURE 6. **ProGAN architecture [4].**

is a trick used in the paper to make the data manifold more continuous and thus more conducive to later training. They also add the Kullback-Leibler divergence of the input Gaussian distribution and the Gaussian distribution as a regularisation term to the Generator's training output, to make the data manifold more continuous and training-friendly.

The Stage I GAN uses the following objective function:

$$\mathcal{L}_{D_0} = \mathbb{E}_{(I_0,t)\sim p_{\text{data}}}\left[\log D_0\left(I_0, \phi_t\right)\right] + \mathbb{E}_{z\sim p_z, t\sim p_{\text{data}}}\left[\log\left(1 - D_0\left(G_0\left(z, \hat{c}_0\right), \phi_t\right)\right)\right] \quad (9)$$

$$\mathcal{L}_{G_0} = \mathbb{E}_{z\sim p_z, t\sim p_{\text{data}}}\left[\log\left(1 - D_0\left(G_9\left(z, \hat{c}_0\right), \phi_t\right)\right)\right] + \lambda D_{KL}\left(\mathcal{N}\left(\mu_0(\phi_t), \Sigma_0(\phi_t)\right) \| \mathcal{N}(0, I)\right) \quad (10)$$

The Stage II GAN uses the following objective function:

$$\mathcal{L}_D = \mathbb{E}_{(I,t)\sim p_{\text{data}}}\left[\log D(I, \phi_t)\right] + \mathbb{E}_{s_0\sim p_{G_0}, t\sim p_{\text{data}}}\left[\log\left(1 - D(G(s_0, \hat{c}), \phi_t)\right)\right] \quad (11)$$

$$\mathcal{L}_G = \mathbb{E}_{s_0\sim p_{G_0}, t\sim p_{\text{data}}}\left[\log\left(1 - D(G(s_0, \hat{c}), \phi_t)\right)\right] + \lambda D_{KL}\left(\mathcal{N}(\mu(\phi_t), \Sigma(\phi_t)) \| \mathcal{N}(0, I)\right) \quad (12)$$

where ϕ_t is the text embedding of the given description, p_z is Gaussian distribution, $\hat{c}_0$ is sampled from a Gaussian distribution from which ϕ_t is drawn. $s_0 = G_0(z, \hat{c}_0)$ and $\lambda = 1$.

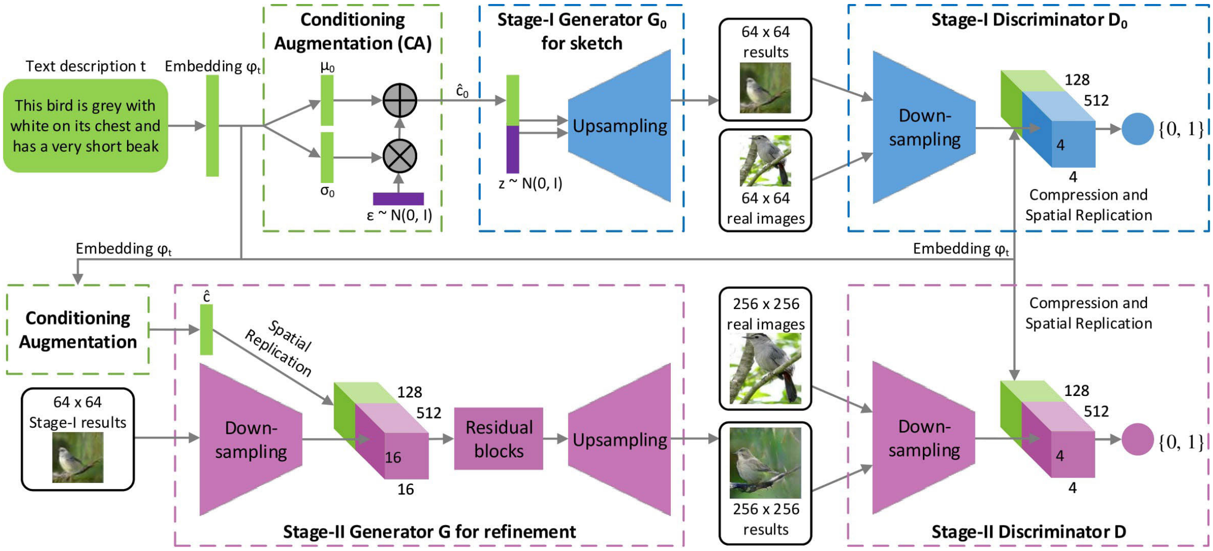


FIGURE 7. StackGAN architecture [22].

H. IMAGE-TO-IMAGE TRANSLATION WITH CONDITIONAL ADVERSARIAL NETWORKS (PIX2PIX)

pix2pix [6] is a conditional generative adversarial network(cGAN [18]) for solving general purpose image-to-image translation problems. The GAN consists of a Generator which has a U-Net [23] architecture and the Discriminator is a PatchGAN [6] classifier. The pix2pix model not only learns the mapping from input to output image, but also constructs a loss function to train this mapping. Interestingly, unlike regular GANs, there is no random noise vector input to the pix2pix Generator. Instead, the Generator learns a mapping from the input image x to the output image $G(x)$. The objective or the loss function for the Discriminator is the traditional adversarial loss function. The Generator on the other hand is trained using the adversarial loss along with the L_1 or pixel distance loss between the generated image and the real or target image. The L_1 loss encourages the generated image for a particular input to remain as similar as possible to the corresponding output real or ground truth image. This leads to faster convergence and more stable training. The loss function for conditional GAN is given by

$$\begin{aligned}\mathcal{L}_{cGAN}(G, D) = &\ \mathbb{E}_{x,y}[\log D(x, y)] \\ &+ \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))]\end{aligned} \tag{13}$$

The L_1 or pixel distance loss is given by

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}\left[\|y - G(x, z)\|_1\right] \tag{14}$$

The final loss function is given by

$$\arg\min_{G}\max_{D} \mathcal{L}_{cGAN}(G, D) + \lambda\mathcal{L}_{L1}(G) \tag{15}$$

where λ is the weighting hyper-parameter coefficient. Pix2PixHD [24] is an improved version of the Pix2Pix algorithm. The primary goal of Pix2PixHD is to produce high-resolution images and perform semantic manipulation. To do this the authors introduced multi-scale Generators and Discriminators and combined the cGANs and feature matching loss function. The training set consists of pairs of corresponding images (s_i, x_i, where s_i is a semantic label map and x_i is a corresponding natural image. The cGAN loss function is given by,

$$\mathbb{E}_{(\mathbf{s},\mathbf{x})}[\log D(\mathbf{s}, \mathbf{x})] + \mathbb{E}_{\mathbf{s}}[\log(1 - D(\mathbf{s}, G(\mathbf{s}))] \tag{16}$$

The ith-layer feature extractor of Discriminator D_k as ${D_k}^{(i)}$(from input to the ith layer of D_k). The feature matching loss $\mathcal{L}_{FM}(G, D_k)$ is given by

$$\begin{aligned}\mathcal{L}_{\mathrm{FM}}(G, D_k) = \mathbb{E}_{(\mathbf{s},\mathbf{x})} &\sum_{i=1}^{T} \frac{1}{N_i} \\ &\left[\left\|D_k^{(i)}(\mathbf{s}, \mathbf{x}) - D_k^{(i)}(\mathbf{s}, G(\mathbf{s}))\right\|_1\right]\end{aligned} \tag{17}$$

where T is the total number of layers and N_i denotes the number of elements in each layer. The objective function of pix2pixHD is given by

$$\begin{aligned}\min_{G}&\left(\max_{D_1,D_2,D_3} \sum_{k=1,2,3} \mathcal{L}_{\mathrm{GAN}}(G, D_k)\right) \\ &+ \lambda \sum_{k=1,2,3} \mathcal{L}_{\mathrm{FM}}(G, D_k)\end{aligned} \tag{18}$$

I. UNPAIRED IMAGE-TO-IMAGE TRANSLATION USING CYCLE-CONSISTENT ADVERSARIAL NETWORKS(CycleGAN)

One fatal flaw of pix2pix is that it requires paired images for training and thus cannot be used for unpaired data which do not have input and output pairs. CycleGAN [7] addresses the issue by introducing a cycle consistency loss that tries to preserve the original image after a cycle of translation and

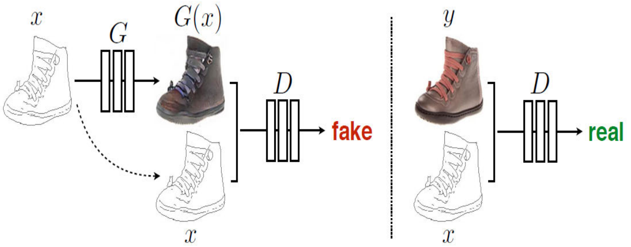


FIGURE 8. Using pix2pix to map edges to color images [6]. D, the Discriminator, learns to distinguish between fake (Generator-generated) and actual (edge, photo) tuples. G, the Generator, learns how to deceive the Discriminator. In contrast to an unconditional GAN, the Generator and Discriminator both look at the input edge map.

reverse translation. Matching pairs of images are no longer required for training in this formulation. CycleGAN uses two Generators and two Discriminators. The Generator G is used to convert images from the X to the Y domain. The Generator F, on the other hand, converts images from Y to X. ($G : X \to Y, F : Y \to X$). The Discriminator D_Y distinguishes y from $G(x)$ and the Discriminator D_X distinguishes x from $F(y)$. The adversarial loss is applied to both the mapping functions. For the mapping function $G : X \to Y$ and its Discriminator D_Y, the objective function is given by

$$\mathcal{L}_{\text{GAN}} (G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}} (y)} \left[\log D_Y(y)\right] + \mathbb{E}_{x \sim p_{\text{data}} (x)} \left[\log (1 - D_Y(G(x))\right] \tag{19}$$

The authors argued that the adversarial losses alone cannot guarantee that the learned function can map an individual input x_i to a desired output y_i as it leaves the model underconstrained. The authors therefore used the cycle consistency loss such that the learned mapping is cycle-consistent. It is based on the assumption that if you convert an image from one domain to the other and back again by feeding it through both Generators in sequence, you should get something similar to what you put in. Forward cycle consistency is represented as $x \to G(x) \to F(G(x)) \approx x$ and the backward cycle consistency as $y \to F(y) \to G(F(y)) \approx y$. The cycle consistency loss is given by

$$\mathcal{L}_{\text{cyc}} (G, F) = \mathbb{E}_{x \sim p_{\text{data}} (x)} [\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}} (y)} [\|G(F(y)) - y\|_1] \tag{20}$$

The final full objective is given by

$$\mathcal{L} (G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}} (G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}} (F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F), \tag{21}$$

where λ controls the relative importance of the two objectives.

$$G^*, F^* = \arg \min_{G,F} \max_{D_x,D_Y} \mathcal{L} (G, F, D_X, D_Y) \tag{22}$$

J. A STYLE-BASED GENERATOR ARCHITECTURE FOR GENERATIVE ADVERSARIAL NETWORKS(StyleGAN)

The primary goal of StyleGAN [10] is to produce high quality, high resolution facial images that are diverse in nature and provide control over the style of generated synthetic images. StyleGAN is an extension of the ProGAN [4] model which uses the progressive growing approach for synthesizing high resolution and high quality images via the incremental (gradual) growing of the Discriminator and the Generator networks during the training process. It's important to note that StyleGAN changes affect only the Generator network, which means they only affect the generative process. The Discriminator and loss function, which are both the same as in a traditional GAN, have not been altered. The upgraded Generator includes several additions to the ProGAN's Generators which are described below:

- **Baseline Progressive GAN:** The authors use the Progressive GAN(ProGAN [4]) as their baseline from which they inherit the network architecture and some of the hyperparameters.
- **Bi-linear up/down sampling:** The ProGAN model used the nearest neighbor up/down sampling but the authors of StyleGAN used bi-linear sampling layers for both the Generator and the Discriminator.
- **Mapping Network, Style Network and AdaIN:** Instead of feeding in the noise vector z directly into the Generator, it goes through a mapping network to get an intermediate noise vector w, say. The output of the mapping network (w) is then passed through a learned affine transformation (A) before passing into the synthesis network through the Adaptive Instance Normalization [25] or AdaIN module. In Figure "A" stands for a learned affine transform. The AdaIN module transfers the encoded information, created by the Mapping Network after the affine transformation, which is incorporated into each block of the Generator model after the convolutional layers. The AdaIN module begins by converting the output of the feature map to a standard Gaussian and then adding the style vector as a bias term. The mapping network f is a standard deep neural network which is comprised of 8 fully connected layers and the synthesis network g consists of 18 layers.
- **Remove traditional input:** Most models, including ProGAN, utilize random input to generate the Generator's initial image. However, the StyleGAN authors found that the image features are controlled by w and the AdaIN. As a result, they simplify the architecture by eliminating the traditional input layer and begin image synthesis with a learned constant tensor.
- **Add noise inputs:** Before evaluating the nonlinearity, Gaussian noise is added after each convolution. In Figure 7. "B" is the learned scaling factor applied per channel to the noise input.
- **Mixing regularization:** The authors also introduced a novel regularization method to reduce neighbouring styles correlation and have more fine grained control

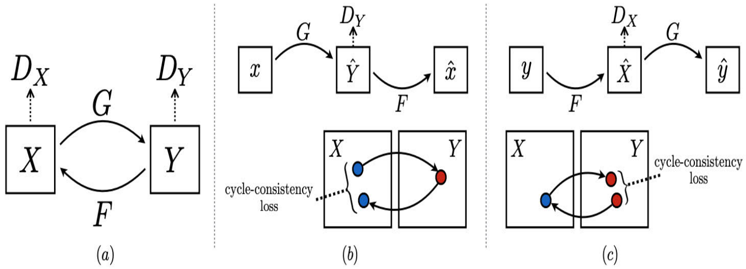


FIGURE 9. CycleGAN [7] (a) two Generator mapping functions $G : X \to Y$ and $F : Y \to X$, and two Discriminators D_Y and D_X. (b) forward cycle-consistency loss: $x \to G(x) \to F(G(x)) \approx x$ and (c) backward cycle consistency loss: $y \to F(y) \to G(F(y)) \approx y$.

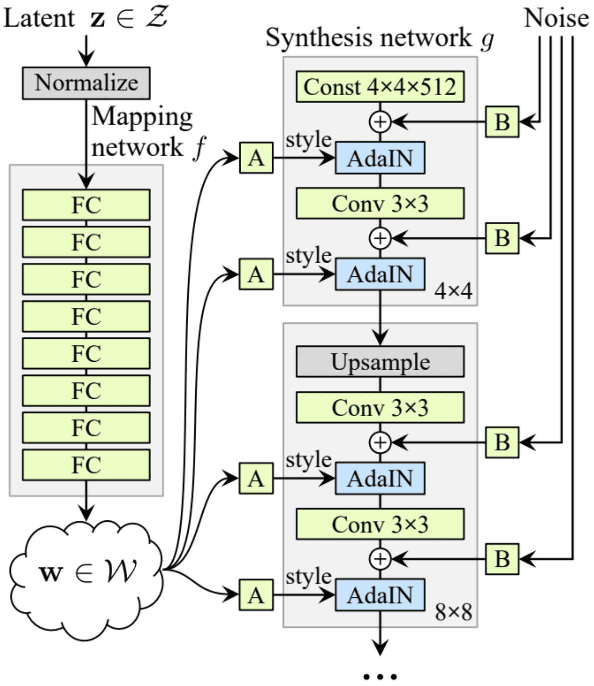


FIGURE 10. StyleGAN generator [10].

over the generated images. Instead of passing just one latent vector, z, through the mapping network as input and getting one vector, w, as output, mixing regularisation passes two latent vectors, z_1 and z_2, through the mapping vector and gets two vectors, w_1 and w_2. The use of w_1 and w_2 is completely randomized for every iteration this technique prevents the network from assuming that styles adjacent to each other correlate.

K. RECURRENT GAN (RGAN) AND RECURRENT CONDITIONAL GAN (RCGAN)

Besides generating synthetic images, GAN can also generate sequential data [26], [27]. Instead of modeling the data distribution in the original feature space, the generative model for time-series data also captures the conditional distribution $P(X_t|X_{1:t-1})$ given historical data. The main difference in architecture between Recurrent GAN and the traditional GAN is that we replace the DNNs/ CNNs with Recurrent Neural Networks (RNNs) in both Generator and Discriminator. Here, the RNNs can be any variants of RNN, such as Long short-term memory (LSTM) and Gated Recurrent Unit (GRU), which captures the temporal dependency in input data. In the case of Recurrent Conditional GAN (RCGAN), both Generator and Discriminator are conditioned on some auxiliary information. Many experiments [27] show that RGAN and RCGAN are able to effectively generate realistic time-series synthetic data.

For RGAN-and-RCGAN, the Generator RNN takes the random noise at each time step to generate the synthetic sequence. Then, the Discriminator RNN works as a classifier to distinguish whether the input is real or fake. Condition inputs are concatenated to the sequential inputs of both the Generator and Discriminator if it is an RCGAN. Similar to GAN, the Discriminator in RGAN minimize the cross-entropy loss between the generated data and the real data. The Discriminator loss is formulated as follows.

$$D_{loss}(X_n, y_n) = -CE(RNN_D(X_n), y_n) \tag{23}$$

where X_n ($X_n \in \mathbb{R}^{T\times d}$) and y_n ($y_n \in \{1, 0\}^T$) are the input and output of the Discriminator with sequence length T and feature size d. y_n is a vector of 1s for real sequence and 0s for synthetic sequence. $CE(\cdot)$ is the average cross-entropy function and $RNN_D(\cdot)$ is the RNN in Discriminator. The Generator loss is formulated below.

$$\begin{aligned} G_{\text{loss}}(Z_n) &= D_{\text{loss}}(\text{RNN}_G(Z_n), 1) \\ &= -CE(\text{RNN}_D(\text{RNN}_G(Z_n)), 1) \end{aligned} \tag{24}$$

Here, Z_n is a random noise vector with $Z_n \in \mathbb{R}^{T\times m}$. In the case of RCGAN, the inputs of both Generator and Discriminator also concatenate the conditional information c_n at each time step.

L. TIME-SERIES GAN (TimeGAN)

Recently, a novel GAN framework preserving temporal dynamics called Time-Series GAN (TimeGAN) [28] is

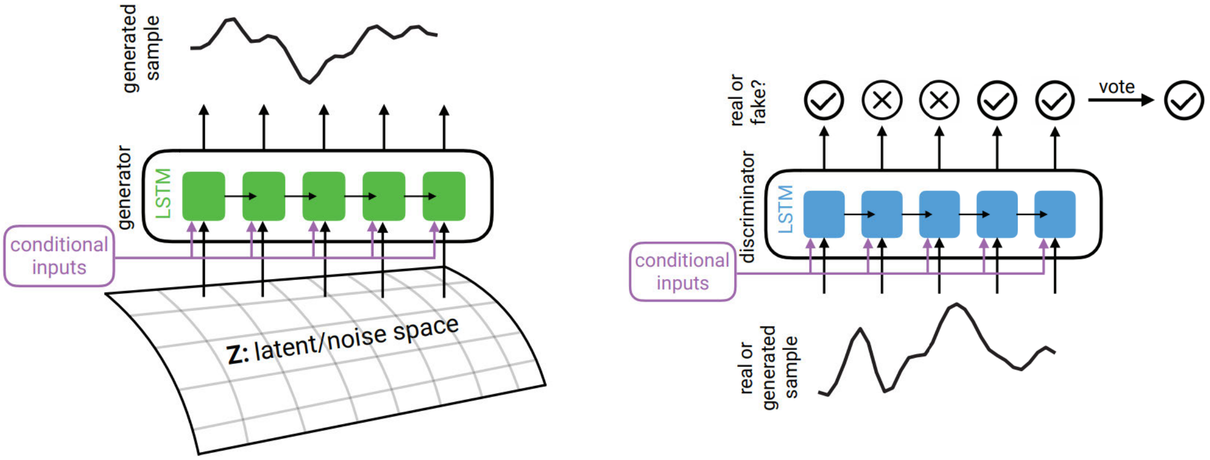


FIGURE 11. RGAN and RCGAN [27].

proposed. Besides minimizing the unsupervised adversarial loss in the traditional GAN learning procedure, (1) TimeGAN introduces a stepwise supervised loss using the original inputs as supervision, which explicitly encourages the model to capture the stepwise conditional distributions in the data. (2) TimeGAN introduces an autoencoder network to learn the mapping from feature space and embedding/latent space, which reduces the dimensionality of the adversarial learning space. (3) To minimize the supervised loss, jointly training on both the autoencoder and Generator is deployed, which forces the model to be conditioned on the embedding to learn temporal relationship. TimeGAN framework not only captures the distributions of features in each time step but also capture the complex temporal dynamics of features across time.

The TimeGAN consists of four important parts, embedding function, recovery function, Generator and Discriminator. First, the autoencoder (first two parts) learns the latent representation from the inputs sequence. Then, the adversarial model (latter two parts) trains jointly on the latent space to generate the synthetic sequence with temporal dynamics by minimizing both the unsupervised loss and supervised loss.

III. GAN EVALUATION METRICS

One of the most difficult aspects of GAN training is assessing their performance, or determining how well a model approximates a data distribution. In terms of theory and applications, significant progress has been made, with a large number of GAN variants now available. However there has been relatively little effort put into evaluating GANs, and there are still gaps in quantitative assessment methods. In this section, we present the relevant and popular metrics which are used to evaluate the performance of GANs.

1) **Inception Score (IS):** IS was proposed by Salimans et al. [29] and it employs the pre-trained Inception-Net [30] trained on ImageNet [31] to capture the desired properties of generated samples. The average Kullback-Leibler or KL divergence [32] between the conditional label distribution $p(y \mid \mathbf{x})$ of samples and the marginal distribution $p(y)$ calculated from all samples is measured by IS. The goal of IS is to assess two characteristics for a set of generated images: image quality (which evaluates whether images have meaningful objects in them) and image diversity. Thus IS favors a low entropy of $p(y \mid \mathbf{x})$ but a high entropy of $p(y)$. IS can be expressed as:

$$\exp\left(\mathbb{E}_{\mathbf{x}}[\mathbb{KL}(p(\mathrm{y} \mid \mathbf{x}) \| p(\mathrm{y}))]\right) \tag{25}$$

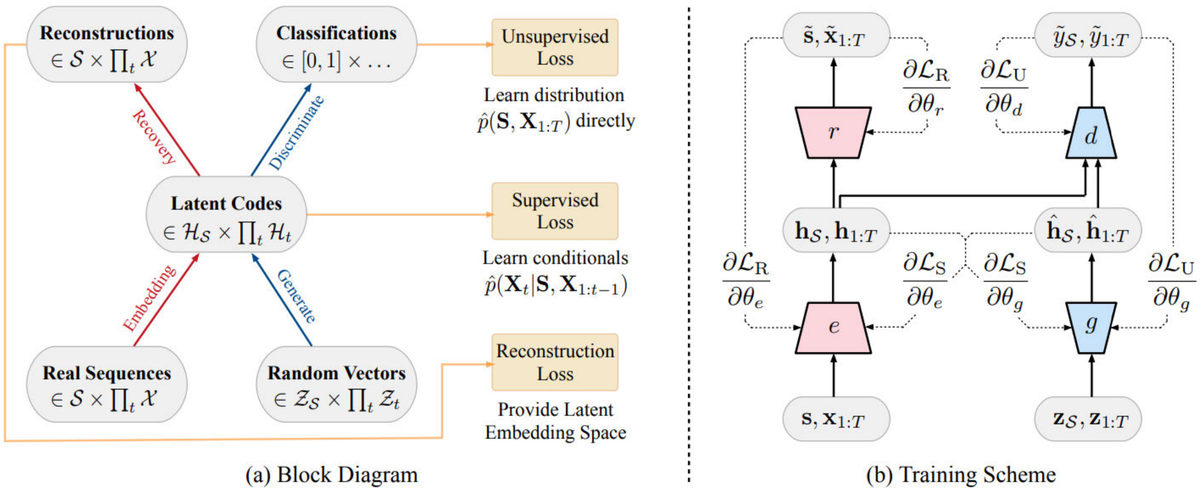
A higher IS indicates that the generative model is capable of producing high-quality samples that are also diverse.

2) **Modified Inception Score (m-IS):** In its original form, Inception Score assigns models that produce a low entropy class conditional distribution $p(y \mid \mathbf{x})$ with a higher score overall generated data. It is, however, desirable to have diversity within a category of samples. To characterize this diversity, Gurumurthy et al. [33] proposed a modified version of inception-score which incorporates a cross-entropy style score $-p\left(y \mid \mathbf{x}_i\right) \log\left(p\left(y \mid \mathbf{x}_j\right)\right)$ where $x_j s$ are samples of the same class as x_i as per the outputs of the trained inception model. The modified IS can be defined as,

$$\exp\left(\mathbb{E}_{\mathbf{x}_i}\left[\mathbb{E}_{\mathbf{x}_j}\left[\left(\mathbb{KL}\left(P\left(y \mid \mathbf{x}_i\right) \| P\left(y \mid \mathbf{x}_j\right)\right)\right]\right]\right) \tag{26}$$

The m-IS is calculated per-class and then averaged across all classes. m-IS can be thought of as a proxy for assessing both intra-class sample diversity and sample quality.

3) **Mode Score(MS):** The MODE score introduced by Che et al. [34] is an improved version of the IS that addresses one of the IS's major flaws: it ignores the prior distribution of ground truth labels. In contrast to IS, MS can measure the difference between the real and


Reconstructions
$\in \mathcal{S} \times \prod_t \mathcal{X}$
Classifications
$\in [0,1] \times \ldots$
Unsupervised Loss
Learn distribution
$\hat{p}(\mathbf{S}, \mathbf{X}_{1:T})$ directly
Recovery
Discriminate
Latent Codes
$\in \mathcal{H}_{\mathcal{S}} \times \prod_t \mathcal{H}_t$
Supervised Loss
Learn conditionals
$\hat{p}(\mathbf{X}_t | \mathbf{S}, \mathbf{X}_{1:t-1})$
Embedding
Generate
Real Sequences
$\in \mathcal{S} \times \prod_t \mathcal{X}$
Random Vectors
$\in \mathcal{Z}_{\mathcal{S}} \times \prod_t \mathcal{Z}_t$
Reconstruction Loss
Provide Latent Embedding Space
$\tilde{\mathbf{s}}, \tilde{\mathbf{x}}_{1:T}$
$\tilde{y}_{\mathcal{S}}, \tilde{y}_{1:T}$
$\frac{\partial \mathcal{L}_{\mathrm{R}}}{\partial \theta_r}$
$\frac{\partial \mathcal{L}_{\mathrm{U}}}{\partial \theta_d}$
r
d
$\mathbf{h}_{\mathcal{S}}, \mathbf{h}_{1:T}$
$\hat{\mathbf{h}}_{\mathcal{S}}, \hat{\mathbf{h}}_{1:T}$
$\frac{\partial \mathcal{L}_{\mathrm{R}}}{\partial \theta_e}$
$\frac{\partial \mathcal{L}_{\mathrm{S}}}{\partial \theta_e}$
$\frac{\partial \mathcal{L}_{\mathrm{S}}}{\partial \theta_g}$
$\frac{\partial \mathcal{L}_{\mathrm{U}}}{\partial \theta_g}$
e
g
$\mathbf{s}, \mathbf{x}_{1:T}$
$\mathbf{z}_{\mathcal{S}}, \mathbf{z}_{1:T}$

(a) Block Diagram

(b) Training Scheme

FIGURE 12. (a) Block Diagram of Four Key Components in TimeGAN. (b) Training scheme: solid lines and dashed lines represent forward propagation paths and backpropagation paths respectively [28].

TABLE 1. Application of common GAN models.

Common GAN variants and extensions	Application area
DCGAN [5]	Image Generation
cGAN [18]	Semi supervised conditional Image Generation
StackGAN [22]	Image generation based on text inputs
Pix2Pix [6]	Image to image translation
CycleGAN [7]	Unpaired Image to image translation
RGAN, RCGAN [27]	Synthetic medical time series data generation
TimeGAN [28]	Realistic time-series data generation

generated distributions.

$$\exp\left(\mathbb{E}_{\mathbf{x}}\left[\mathbb{KL}\left(p(y \mid \mathbf{x}) \| p\left(y^{\text{train}}\right)\right)\right] - \mathbb{KL}\left(p(y) \| p\left(y^{\text{train}}\right)\right)\right) \tag{27}$$

where $p\left(\mathrm{y}^{\text{train}}\right)$ is the empirical distribution of labels computed from training data. According to the author's evaluation, the MODE score successfully measures two important aspects of generative models, namely variety and visual quality.

4) **Frechet Inception Distance (FID):** Proposed by Heusel et al. [35], the Frechet Inception Distance score determines how far feature vectors calculated for real and generated images differ. A specific layer of the InceptionNet [30] model is used by the FID score to capture and embed features of an input image. The embeddings are summarized as a multivariate Gaussian by calculating the mean and covariance for both the generated data and the real data. The Fréchet distance (or Wasserstein-2 distance) between these two Gaussians is then used to quantify the quality of the generated samples.

$$FID(r, g) = \left\|\mu_r - \mu_g\right\|_2^2 + \operatorname{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{\frac{1}{2}}) \tag{28}$$

where $\left(\mu_g, \Sigma_g\right)$ and $\left(\mu_r, \Sigma_r\right)$ represent the empirical mean and empirical covariance of the generated and real data disctibutions respectively. Smaller distances between synthetic(model generated) and real data distributions are indicated by a lower FID.

5) **Image Quality Measures:** Below we describe some commonly used image quality assessment measures used to compare GAN generated data with the real data.
   a) Structural similarity index measure (SSIM) [36] is a method for quantifying the similarity between two images. SSIM tries to model the perceived change in the image's structural information. The SSIM value varies between -1 and 1, where a value of 1 shows perfect similarity. Multi-Scale SSIM or MS-SSIM [37] is a multiscale version of SSIM that allows for more flexibility in incorporating image resolution and viewing conditions than a single scale approach. MS-SSIM ranges between 0 (low similarity) and 1 (high similarity).
   b) Peak Signal-to-Noise Ratio or PSNR compares the quality of a generated image to its corresponding real image by measuring the peak signal-to-noise ratio of two monochromatic images. For example evaluation of conditional GANs or cGANs [18] Higher PSNR (in db) indicates better quality of the generated image.

c) Sharpness Difference (SD) represents the difference in clarity between the generated and the real image. The larger the value is, the smaller the difference in sharpness between the images is and the closer the generated image is to the real image.

6) **Evaluation Metrics for Time-Series/Sequence Inputs:** To evaluate quality of generating synthetic sequence data is very challenging. For example, the Intensive Care Unit (ICU) signal looks completely random to a non-medical export [27]. The researchers evaluate the quality of synthetic sequential data mainly focusing on the following three different aspects. (1) Diversity – the synthetic data should be generated from the same distribution of real data. (2) Fidelity – the synthetic data should be indistinguishable from the real data. (3) Usability — the synthetic data should be good enough to be used as the train/test dataset. [28]

a) t-SNE and PCA [28] are both commonly used visualization tools for analyzing both the original and synthetic sequence datasets. They flatten the dataset across the temporal dimension so that the dataset can be plotted in the 2D plane. They measure how closely the distribution of generated samples resembles that of the original in 2-dimensional space.

b) Discriminative Score [28] evaluates how difficult for a binary classifier to distinguish between the real (original) dataset and the fake (generated) dataset. It is challenging for the classifier to classify if the synthetic data and the original are drawn from the same distribution.

c) Maximum Mean Discrepancy (MMD) [27], [38] learns the distribution of the real data. The maximum mean discrepancy method has been proposed to distinguish whether the synthetic data and the real data are from the same distribution by computing the squared difference of the statistics between real and synthetic samples (MMD^2). The unbiased MMD^2 can be denoted as following where the inner production between functions is replaced with a kernel function K.

$$\begin{aligned}\widehat{MMD}^2 = &\frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i}^{n}K(x_i, x_j)\\ &-\frac{2}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m}K(x_i, y_j)\\ &+\frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{j\neq i}^{m}K(y_i, y_j) \quad (29)\end{aligned}$$

A suitable kernel function for the time-series data is vital. The authors treat the time series as vectors for comparison and select the radial basis function (RBF) as the kernel function which is $K(x, y) = exp(-\|x-y\|^2/(2\sigma^2))$. To select an appropriate kernel bandwidth σ, the estimator of the t-statistic of the power of the MMD test between two distributions $\widehat{t} = \frac{\widehat{MMD}^2}{\sqrt{\widehat{V}}}$ is maximised. The authors split a validation set during training to tune the parameter. The result shows that MMD^2 is more informative than either Generator or Discriminator loss, and correlates well with quality as assessed by visualising [27].

d) Earth Mover Distance (EMD) [39], [40] is a measure of the distance between two probability distributions over a region. It describes how much probability mass has to be moved to transform P^h into P^g where P^h denotes the historical distribution and P^g is the generated distribution. The EMD is defined by

$$\begin{aligned}EMD(P^h, P^g) = &\inf_{\pi\in\prod(P^h,P^g)}\\ &E_{(X,Y)\sim\pi}\left[\|X-Y\|\right] \quad (30)\end{aligned}$$

where $\prod(P^h, P^g)$ denotes the set of all joint probability distributions with marginals P^h and P^g.

e) AutoCorrelation Function (ACF) Score [39] describes the coefficient of correlation between historical and the generated time series. Let $r_{1:T}$ denote the historical log percentage change series and $\{r^{(1)}_{1:\widetilde{T},\theta}, \ldots, r^{(M)}_{1:\widetilde{T},\theta}\}$ a set of generated log percentage change paths of length $\widetilde{T} \in N$. The autocorrelation is calculated with the time lag *tau* and the series $r_{1:T}$ and measures the correlation of the lagged time series with the series itself

$$C(\tau; r) = Corr(r_{t+\tau}, r_t) \quad (31)$$

The ACF(f) score is computed for a function $f : R \to R$ as

$$\begin{aligned}ACF(f) := &\| C(f(r_{1:T}))\\ &-\frac{1}{M}\sum_{i=1}^{M}C(f(r^{(i)}_{1:T,\theta}))\|_2 \quad (32)\end{aligned}$$

where $C : R^T \to [-1, 1]^S : r_{1:T} \mapsto (C(1; r), \ldots, C(S; r))$.

f) Leverage Effect Score [39] provides a comparison of the historical and the generated time dependence. The leverage effect for lag τ is measured using the correlation of the lagged, squared log percentage changes and the log percentage changes themselves.

$$\ell(\tau; r) = Corr(r^2_{t+\tau}, r_t) \quad (33)$$

The leverage effect score is defined by

$$\| L(r_{1:T} = \frac{1}{M}\sum_{i=1}^{M}L(r^{(i)}_{1:\widetilde{T},\theta}))\|_2 \quad (34)$$

where $L : R^T \to [-1, 1]^S : r_{1;r} \mapsto (\ell(1; r), \ldots, \ell(S; r))$.

TABLE 2. Summary of relevant and popular GAN evaluation metrics.

Quantitative metrics	Description
Inception Score (IS) [29]	Measures the KL-Divergence between the conditional and marginal label distributions over the data.
Modified Inception Score (m-IS) [33]	Incorporates a cross-entropy style score to promote diversity within a category of samples.
Mode Score (MS) [34]	Improved veersion of IS and takes into account the prior distribution of ground truth labels.
Fréchet Inception Distance (FID) [35]	Evaluates the Fréchet distance or the Wasserstein-2 distance between the multi-variate Gaussians fitted to data embedded into a feature space.
Structural Similarity Index Measure (SSIM [36]), Peak signal-to-noise ratio (PSNR), Multiscale SSIM (MS-SSIM [37]) and Sharpness Difference (SD)	Evaluate and assess the quality of generated images.
Maximum Mean Discrepancy (MMD [26], [38]), Earth Mover Distance (EMD [39], [40]), DY Metric [39], ACF Score [39], Leverage Effect Score [39]	Evaluate the quality of generated sequence data.

IV. GAN APPLICATIONS

GANs are by far the most widely used generative models and they are immensely powerful for the generation of realistic synthetic data samples. In this section, we will go over the wide array of domains in which Generative Adversarial Networks (GANs) are being applied. Specifically, we will present the use of GANs in Image processing, Video generation and prediction, Medical and Healthcare, Biology, Astronomy, Remote Sensing, Material Science, Finance, Fashion Design, Sports and Music.

A. IMAGE PROCESSING

GANs are quite prolific when it comes to specific image processing related tasks like image super-resolution, image editing, high resolution face generation, facial attribute manipulation to name a few.

- **Image super-resolution:** Image super-resolution refers to the process of transforming low resolution images to high resolution images. SRGAN [8] is the first image super-resolution framework capable of inferring photo-realistic natural images for 4x up-scaling factors. Several other super resolution frameworks([9], [11]) have also been developed to produce better results. Best-Buddy GANs [41] developed recently is used for single image super-resolution (SISR) task along with previous works([42], [43]).
- **Image editing:** Image editing involves removing or modifying some aspects of an image. For example, images captured during bad weather or heavy rain lack visual quality and thus will require manual intervention to either remove anomalies such as raindrops or dust particles that reduce image quality. The authors of ID-CGAN [44] used GANs to address the problem of single image de-raining. Image modification could involve modifying or changing some aspects of an image such as changing the hair color, adding a smile, etc. which was demonstrated by ([45], [46]).
- **High resolution face image generation:** High resolution facial image generation is one other area of image processing where GANs have excelled. Face generation and attribute manipulation using GANs can be broadly classified into the creation of entire synthetic faces, face features or attribute manipulation and face component transformation.
  - **Synthetic face generation:** Synthetic face generation refers to the creation of synthetic images of the face of people who do not exist in real life. ProGAN [4], described in the previous section demonstrated the generation of realistic looking images of human faces. Since then there have been several works which use GANs for facial image generation([47], [48]). StyleGAN [49] which is a unique generative adversarial network introduced by Nvidia researchers in December 2018. The primary goal of StyleGAN is to generate high quality face images that are also diverse in nature. To achieve this, the authors used techniques such as using a noise mapping network, adaptive instance normalization and progressive growing similar to ProGAN to produce very high resolution images.
  - **Face features or attribute manipulation:** Face attribute manipulation includes facial pose and expression transformation. The authors of PosIX-GAN [50] trained their model to generate high quality face images with 9 different pose variations when given a face image in any arbitrary pose as input. DECGAN [51] authors used Double Encoder Conditional GAN to perform facial expression synthesis. Expression Conditional GAN (ECGAN) [52] can learn the mapping from one image domain to another and the authors were able to control specific facial expressions by the conditional attribute label.
  - **Face component transformation:** Face component transformation deals with altering the face style(hair color and style) or adding accessories such as eye glasses. The authors of DiscoGAN [53] were able to change hair color and the authors of StarGAN [54] were able to perform multi-domain image translations. BeautyGAN [55] can be used to translate the makeup style from a given reference makeup face image to another non-makeup one

while preserving face identity. The authors of InfoGAN [21] trained their model to learn disentangled representations in an unsupervised manner and can modify facial components such as adding or removing eyeglasses and changing hairstyles. GANs can also be applied to image inpainting where the task is of reconstructing missing regions in an image. In this regard GANs have been used([12], [56]) to perform the task.

B. VIDEO GENERATION AND PREDICTION

Synthesizing videos using GANs can be divided into three main categories.

- Unconditional video generation
- Conditional video generation
- Video prediction

1) UNCONDITIONAL VIDEO GENERATION

For **unconditional video generation** the output of the GAN models is not conditioned on any input signals. Due to the lack of any information provided as a condition with the videos during the training phase, the output videos produced by these frameworks typically are low-quality in nature.

The authors of VGAN [57] were the first to apply GANs for video generation. Their Generator consists of two CNN networks, one 3D spatio-temporal convolutional network to capture moving objects in the foreground, and the other is a 2D spatial convolutional model that captures the static background. The two independent outputs from the Generator are combined to create the generated video and fed to the Discriminator, to decide if it is real or fake. Temporal Generative Adversarial Nets (TGAN) [58] can learn representation from an unlabeled video dataset and generate a new video. TGAN Generator consists of two sub Generators one of which is the temporal Generator and the other is the image Generator. The temporal Generator takes a single latent variable as input and produces a set of latent variables, each of which corresponds to a video frame. The image Generator creates a video from a set of latent variables. The Discriminator consists of three-dimensional convolutional layers. TGAN uses WGAN to provide stable training and meets the K-Lipschitz constraint. FTGAN [59] consists of two GANs: FlowGAN and TextureGAN. FlowGAN network deals with motion, i.e. adds optical flow for representing the object motion more effectively. The TextureGAN model is used to generate the texture that is conditioned on the previous FlowGAN result, to produce the required frames. Motion and Content decomposed Generative Adversarial Network or MoCoGAN [60] uses a motion and content decomposed representation for video generation. MoCoGAN is made up of four sub-networks: a recurrent neural network, an image Generator, an image Discriminator, a video Discriminator, and a video Discriminator. Built on the BigGAN [61] architecture, the Dual Video Discriminator GAN (DVD-GAN) [62] is a generative video model for high quality frame generation. DVD-GAN employs Recurrent Neural Network (RNN) units as well as a dual Discriminator architecture to deal with the spatial and temporal dimension.

2) CONDITIONAL VIDEO GENERATION

In **conditional video generation** the output of the GAN models is conditioned on input signals such as text, audio or speech. We do not consider other conditioning techniques such as images to video, semantic map to videos and video to video as these can be considered to fall under the video prediction category which is described in section 4.5.2 below.

For **text to video synthesis** the goal is to generate videos based on some conditional text. Li et al. [63] used Variational Autoencoder (VAE) [64] and Generative Adversarial Network (GAN) for generating videos from text. Their model is made a conditional gist Generator (conditional VAE), a video Generator, and a video Discriminator. The initial image/gist is created by the conditional gist Generator, which is conditioned on encoded text. This gist serves as a general representation of the image, background color and object layout of the desired video. The video's content and motion are then generated using cGAN by conditioning both the gist and the text input. Temporal GANs conditioning on captions (TGANs-C) [58] uses a Bidirectional LSTM and LSTM based encoder to embed and obtain the representation of the input text. This output representation is then concatenated with a random noise vector and then given to the Generator, which is a 3D deconvolution network to generate synthesize realistic videos. The model has three Discriminators: The video Discriminator distinguishes real video from synthetic video and aligns video with the correct caption, the frame Discriminator determines whether each frame is real/fake and semantically matched/mismatched with the given caption, and the motion Discriminator exploits temporal coherence between consecutive frames. Balaji et al. proposed the Text-Filter conditioning Generative Adversarial Network (TFGAN) [65]. TFGAN introduces a novel multi-scale text-conditioning technique in which text features are extracted from the encoded text and used to create convolution filters. Then, the convolution filters are input to the Discriminator network to learn good video-text associations in the GAN model. StoryGAN [66] is based on a sequential conditional GAN framework whose task is to generates a sequence of images for each sentence in a given multi-sentence paragraph. The GAN model consists of (*i*) Story Encoder, (*ii*) A recurrent neural network (RNN) based Context Encoder, (*iii*) An image Generator and (*iv*) An image Discriminator and a story Discriminator. BoGAN [67] maintains semantic matching between video and the corresponding language description at various levels and ensures coherence between consecutive frames. The authors used LSTM and 3D convolution based encoder decoder architecture to produce frames from embedding based on the input text. Region level semantic alignment module was proposed to encourage the Generator to take advantage of the semantic alignment between video and words on a local level. To maintain

TABLE 3. Applications in image processing.

Image processing application	GAN models
Image super-resolution	SRGAN [8], TSRGAN [9], ESRGAN [11], Best-BuddyGAN [41], GMGAN [42], PESR [43]
Image editing and modifying	ID-CGAN [44], IcGAN [45]
Synthetic face generation	ProGAN [4], StyleGAN [10]
Face features or attribute manipulation	PosIX-GAN [50], DECGAN [51], ECGAN [52]
Face component transformation	DiscoGAN [53], StarGAN [54], BeautyGAN [55], InfoGAN [21]

frame-level and video level coherence two Discriminators were used. Kim et al. [68] came up with Text-to-Image-to-Video Generative Adversarial Network (TiVGAN) for text-based video generation. The key idea is to begin by learning text-to-single-image generation, then gradually increase the number of images produced, and repeat until the desired video length is reached.

Speech to video synthesis involves the task of generating synchronized video frames conditioned on an audio/speech input. Jalalifar et al. [69] used LSTM and CGAN for speech conditioned talking face generation. LSTM network learns to extract and predict facial landmark positions from audio features. Given the extracted set of landmarks, the cGAN then generates synchronized facial images with accurate lip sync. Vougioukas et al. [70] used GANs for generating videos of a talking head. The generation of video frames is conditioned on a still image of a person and an audio clip containing speech and does not rely on extracting intermediate features. Their GAN architecture uses an RNN based Generator, frame level and sequence level Discriminator respectively. The idea of disentangled representation was explored by Zhou et al. [71]. The authors proposed the Disentangled Audio-Visual System (DAVS), which uses disentangled audio-visual representation to create high-quality talking face videos.

3) VIDEO PREDICTION

Video prediction is the ability to predict future video frames based on the context of a sequence of previous frames. Formally future frame prediction can be defined as follows. Let $\mathbf{X}_i \in \mathrm{R}^{w \times h \times c}$ be the i^{th} frame in the sequence of n video frames $\mathbf{X} = (X_{i-n}, \ldots, X_{i-1}, X_i)$, where w, h, and c denote the width, height and the number of channels respectively. The goal is to predict the next sequence of frames $\mathbf{Y} = \left(\hat{Y}_{i+1}, \hat{Y}_{i+2}, \ldots, \hat{Y}_{i+m}\right)$ using the input $\mathbf{X}$.

Video prediction is a challenging task due to the complex task of modelling both the content and motion in videos. To this extent several studies have been carried out to perform video prediction using GAN-based training ([72], [73], [74], [75], [76], [77], [78]). Mathieu et al. [72] used multi-scale architecture for future frame prediction. The network was trained using an adversarial training method, and an image gradient difference loss function. MCNet [73] performs the task of video frame prediction by disentangling temporal and spatial dynamics in videos. An Encoder-Decoder Convolutional Neural Network is used to model video content and Convolutional-LSTM is used to model temporal dynamics or motion in the video. In this way predicting the next frame is as simple as converting the extracted content features into the next frame content using the recognized motion features.

FutureGAN [74] used an encoder-decoder based GAN model to predict future frames of the video sequence. Their network comprises of Spatio-temporal 3D convolution network(3D ConvNets) [79] for all encoder and decoder modules to capture both the spatial and temporal components of a video sequence. To have stable training and prevent problems of mode collapse the authors used Wasserstein GAN with gradient penalty (WGAN-GP) [20] loss and the technique of progressively growing GAN or ProGAN [4] which has been shown to generate high resolution images. VPGAN [75] is a GAN-based framework for stochastic video prediction. The authors introduce a new adversarial inference model, an action control conformal mapping network and use cycle consistency loss for their model. The authors also combined image segmentation models with their GAN framework for robust and accurate frame prediction. Their model outperformed other existing stochastic video prediction methods. With a unified architecture, Dual Motion GAN [76] attempts to jointly resolve the future-frame and future-flow prediction problems. The proposed framework takes in as input a sequence of frames to predict the next frame by combining the future frame prediction with the future flow-based prediction. To achieve this, Dual Motion GAN employs a probabilistic motion encoder (to map frames to latent space), two Generators (a future-frame Generator and a future-flow Generator), as well as a flow estimator and flow-warping layer. A frame Discriminator and a flow Discriminator are used to classifying fake and real future frames and flow. Bhattacharjee et al. [80] tackle the problem of future frames prediction by using multi-stage GANs. To capture the temporal dimension and handle inter-frame relationships, the authors proposed two new objective functions, the Normalized Cross-Correlation Loss (NCCL) and the Pairwise Contrastive Divergence Loss (PCDL). The multistage GAN(MS-GAN) is made up of 2 GANs to generate frames at two separate resolution whereby Stage-1 GAN output is fed to Stage-2 GAN to produce the final output. Kwon et al. [81] proposed a novel framework based on CycleGAN [7] called the Retrospective CycleGAN to predict future frames that are farther in time but are relatively sharper than frames generated by other methods. The framework consists of a single Generator and two Discriminators. During training, the Generator is tasked with generating both future and past frames, and the retrospective cycle constraints are used to ensure bi-directional prediction consistency. The frame Discriminator detects individual fake frames, whereas the sequence Discriminator determines whether or not a sequence contains fake frames. According to the authors, the sequence Discriminator plays a crucial role in predicting temporally consistent future frames. To train their model, the combination of two adversarial and two reconstruction loss functions were used.

C. MEDICAL AND HEALTHCARE

GANs have immense medical image generation applications and can be used to improve early diagnosis, reduce time and expenditure. Because medical image data is generally limited, GANs can be employed as data augmentation techniques by conducting image-to-image translation, synthetic data synthesis, and medical image super-resolution.

One major application of GANs in medical and healthcare is the **image-to-image translation** framework, that is when multi-modal images are required one can use images from one modality or domain to generate images in another domain. Magnetic resonance imaging (MRI), is considered the gold standard in medical imaging. Unfortunately, it is not a viable option for patients with metal implants, as the metal in the machine could interfere with the results and the patients' safety. MR-GAN [82] is similar to CycleGAN [7] and is used to transform 2D brain CT image slices into 2D brain MR image slices. However, unlike CycleGAN, which is used for unpaired image-to-image translation, MR-GAN is trained using both paired and unpaired data and combines adversarial loss, dual cycle-consistent loss, and voxel-wise loss. MCML-GANs [83] uses the approach of multiple-channels-multiple-landmarks (MCML) input to synthesize color Retinal fundus images from a combination of vessel tree, optic disc, and optic cup images. The authors used two models based on the pix2pix [6] and CycleGAN [7] model framework and proposed several different architectures for the Generator model and compared their performance. Zhao et al. [84] also proposed Tub-GAN and Tub-sGAN image-to-image translation framework to generate retinal and neuronal images. Armanious et al. [85] proposed the MedGAN framework for generalizing image to image translation in the field of medical image generation by combining the adversarial framework with a new combination of non-adversarial losses along with the usage of CasNet a ResNets [86] inspired architecture. Sandfort et al. [87] used CycleGAN [7] to transform contrast CT images into noncontrast images. The authors compared the segmentation performance of a U-Net trained on the original dataset versus a U-Net trained on a combined dataset of original data and synthetic non-contrast images were compared.

DermGAN [88] is used to generate synthetic images with skin conditions. The model learns to convert a semantic map containing a pre-specified skin condition, its size and location, as well as the underlying skin colour, into a realistic image that retains the pre-specified traits. The DermGAN Generator uses a modified U-Net [23] where the deconvolution layers are replaced with a nearest-neighbor resizing layer followed by a convolution layer to reduce the checkerboard effect. The Generator and Discriminator are trained to minimize the combination of feature matching loss, min-max GAN loss, l_1 reconstruction loss for the whole image, l_1 reconstruction loss for the pathological region.

Apart from solving the image to image translation problems GAN are widely used for **synthetic medical image generation**([89], [90], [91], [92]). Costa et al. [91] implemented an adversarial autoencoder for the task of conditional retinal vessel network synthesis. Beers et al. [93] applied ProGAN to generate high resolution and high quality 512 × 512 retinal fundus images and 256 × 256 multimodal glioma images. Zhang et al. [92] used DCGAN [5], WGAN [19] and boundary equilibrium GANs (BEGANs) [94] to generate synthetic medical images. They used the generated synthetic images to augment their datasets to build models with higher tissue recognition accuracy. Overall training with augmented datasets saw an increase in tissue recognition accuracy for the three GAN models when compared to base-line models trained without data augmentation. fNIRS-GANs [95] based on WGAN [19] is used to perform functional near-infrared spectroscopy (fNIRS) data augmentation to improve the fNIRS-brain- computer interface (BCI) accuracy. Using data augmentation the authors were able to achieve higher classification accuracy of 0.733 and 0.746 for both the SVM and neural network models respectively as compared to 0.4 for both the models trained without data augmentation. To prevent data leakage by generating anonymized synthetic electrocardiograms (ECGs), Piacentino et al. [96] used GANs. The authors first propose a new general procedure to convert raw data into images, which are well suited for GANs. Following that, a GAN design was established, trained, and evaluated. Because of its simplicity, the authors chose to use Auxiliary Classifier Generative Adversarial Network(ACGAN) [97] for their intended task. Kwon et al. [98] used GANs to augment mRNA samples to improve classification accuracy of deep learning models for cancer detection. With 5 fold increase in training data by combining GAN generated synthetic samples with the original dataset, the authors were able to improve the F1 score by 39%.

Image reconstruction and super resolution are vital to obtaining high resolution images for diagnosis as due to constraints such as the amount of radiation used for MRI and other image acquisition techniques can highly impact the quality of images obtained. Multi-level densely connected super-resolution network, mDCSRN-GAN [99] proposed by Chen et al. uses an efficient 3D neural network design for the Generator architecture to perform image super resolution. MedSRGAN [100] is an image super resolution (SR) framework for medical images. The authors used a novel convolutional neural network, Residual Whole Map Attention Network (RWMAN) as the Generator network to

TABLE 4. Applications in video generation and prediction.

Video generation and prediction	GAN models
Unconditional video generation	VGAN [57], TGAN [58], FTGAN [59], MoCoGAN [60], DVD-GAN [62]
Conditional video generation	VAE-GAN [63], TGANs-C [58], TFGAN [65], StoryGAN [66], Bo-GAN [67], TiVGAN [68], LSTM and cGAN [69], RNN based GAN [70], TemporalGAN [71]
Video Prediction	Multi-Scale GAN [72], MCNet [73], FutureGAN [74], VPGAN [75], Dual Motion GAN [76], MSGAN [80], Retrospective Cycle GAN [81]

low resolution features and then performs upsampling. For the Discriminator instead of having just the single generated high resolution image the authors used pairs of input low resolution images and generated high resolution images. Yamashita et. al [101] evaluated several super resolution GAN models(SRCNN [102], VDSR [103], DRCN [104] and ESRGAN [11]) for Optical Coherence Tomography (OCT) [105] image enhancement. The authors found ESRGAN has the worst performance in terms of PSRN and SSIM but qualitatively, it was the best one producing sharper and high contrast images.

D. BIOLOGY

Biology is an area where generative models especially GANs can have a great impact by performing tasks such as protein sequence design, data augmentation and imputation and biological image generation. Apart from this GANs can also be applied for binding affinity prediction.

Protein engineering the process of identifying or developing useful or valuable proteins sequences with certain optimized properties. Several works have been done in relation to the application of Deep Generative models for protein sequence, especially the use of GANs (Repecka et al. [106], Amimeur et al. [107] and Gupta et al. [108]). GANs can be used to generate novel valid functional protein sequences and optimize protein sequences to have certain specific properties. ProteinGAN [106] can learn to generate diverse functional protein sequences directly from complex multidimensional amino acids sequence space. The authors specifically used GANs to generate functional malate dehydrogenases. Amimeur et al. [107] developed the Antibody-GAN which uses a modified WGAN for the generation of both single-chain and paired-chain antibody sequence generation. Their model is capable of generating extremely large diverse libraries of novel libraries that mimic somatically hypermutated human repertoire response. The authors also demonstrated the use of transfer learning to use their GAN model to generate molecules with specific properties of interest like MHC class II binding and specific complementarity-determining region (CDR) characteristics. FBGAN [108] uses the WGAN architecture along with the analyzer in a feedback-loop mechanism to optimize the synthetic gene sequences for desired properties using an external function analyser. The analyzer is a differential neural network and assigns a score to sampled sequences from the Generator. As training progresses lowest scoring generated sequences are replaced by high scoring generated sequences for the entire Discriminator's training set. GANs were utilised by Anand at al. [109] to generate protein structures, with the goal of using them in quick *de novo* protein design.

GANs have been used for **data augmentation and data imputation** in biology due to the lack of available biosamples or the cost of collecting such samples. Some recent works include the generation and analysis of single-cell RNA-seq ([110], [111]). The authors of cscGAN [111] or conditional single-cell generative adversarial neural networks used GANs for the generation of realistic single-cell RNA-seq data. Wang et al. [112] proposed GGAN, a GAN framework to impute the expression values of the unmeasured genes. To do this they used a conditional GAN to leverage the correlations between the set of landmark and target genes in expression data. The Generator takes the landmark gene expression as input and outputs the target gene expression. This approach leverages correlations between the set of landmark and target genes in expression data from projects like 1000 Genomes. Park et al. [113] applied GANs to predict the molecular progress of Alzheimer's disease (AD) by successfully analyzing RNA-seq data from a 5xFAD mouse model of AD. Specifically, the authors successfully applied WGAN+GP [20] to bulk RNA-seq data with fewer variations in gene expression levels and a smaller number of genes. scIGAN [114] is a GAN-based framework for scRNA-seq imputation. scIGANs can use complex, multi-cell type samples to learn non-linear gene-gene correlations and train a generative model to generate realistic expression profiles of defined cell types.

GANs can also be used to **generate biological imaging data**. CytoGAN [115] or Generative Modeling of Cell Images, the authors evaluated the use of several GAN models such as DCGAN [5], LSGAN [116] and WGAN [19] for cell microscopy imaging, in particular morphological profiling. Through their experiments, they discovered that LSGAN was the most stable, resulting in higher-quality images than both DCGAN and WGAN. GANs have also been used for the generation of realistic looking electron microscope images (Han et al. [117]) and the generation of cells imaged by fluorescence microscopy(Oskin et al. [118]).

Predicting **binding affinities** is an important task in drug discovery though it still remains a challenge. To aid in drug discovery by predicting binding affinity between drug and target Zhao et al. [119] devised the use of a semi-supervised

TABLE 5. Applications in medical and healthcare.

Medical and Healthcare application	GAN models
Multimodal Image to image translation	pix2pix [6], CycleGAN [7], MR-GAN [82],MCML-GANs [83], Tub-GAN and Tub-sGAN [84], MedGAN [85], DermGAN [88]
Image generation for data augmentation	WGAN [19], WGAN-GP [20], DCGAN [5], BEGAN [94], ProGAN [4], fNIRS-GANs [95], ACGAN [97]
Image reconstruction	SRCNN [102], VDSR [103], DRCN [104], ESRGAN [11] mDCSRN-GAN [99], MedSRGAN [100]

GANs-based method. The researchers utilised two GANs to learn representations from raw protein and drug sequence data, and a convolutional regression network to predict affinity.

E. ASTRONOMY

With the advent of Big-data, the amount of data publicly available to scientists for data-driven analysis is mind boggling. Every day terabytes of data are being generated by hundreds if not thousands of satellites across the globe. With powerful computing resources GANs have found their way into astronomy as well for tasks such as image translation, data augmentation and spectral denoising.

The authors of RadioGAN [120] based their GAN on the Pix2Pix model to perform **image to image translation** between two different radio survey datasets to recover extended flux densities. Their model recovers extended flux density for nearly half of the sources within a 20% margin of error and learns more complex relationships between sources in the two surveys than simply convolving them with a different synthesised beam. Several other authors have also used image to image translation models such as Pix2Pix, Pix2PixHD to generate solar images(Dash et al., Park et al. [121], Kim et al. [122], Jeong et al. [123], Shin et al. [124] etc.).

Apart from image-to-image related tasks, GANs have been extensively used to **generate synthetic data** in the astronomy domain. Smith et al. [125] proposed SGAN to produce realistic synthetic eXtreme Deep Field(XDF) images similar to the ones taken by the Hubble Space Telescope. Their SGAN model has a similar architecture to DCGAN and can be used to generate synthetic images in astrophysics and other domain. Ullmo et al. [126] used GANs to generate cosmological images to bypass simulations which generally require lots of computing resources and are quite expensive. Dia et al. [127] showed that GANs can replace expensive model-driven approaches to generate astronomical images. In particular, they used ProGANs along with Wasserstein cost function to generate realistic images of galaxies. ExoGAN [128] which is based on the DCGAN framework [5] is the first deep-learning approach for solving the inverse retrieval of exoplanetary atmospheres. According to the authors, ExoGAN was found to be up to 300 times faster than a standard retrieval for large spectral ranges. ExoGAN is designed to work with a wide range of instruments and wavelength ranges without requiring any additional training. Fussell et al. [129] explored the use of DCGAN [5] and StackGAN [22] in a chained fashion for generation of high-resolution synthetic galaxies images.

The authors of Spectra-GAN [130] designed their algorithm for **spectral denoising.** Their algorithm is based on CycleGAN i.e. it has two Generators and two Discriminators, with the exception that instead of unpaired samples SpectraGAN used paired examples. The model comprises of three loss functions: adversarial loss, cycle-consistent loss, and generation-consistent loss.

F. REMOTE SENSING

Using GANs for remote sensing applications can be broadly divided into the following main categories:

- **Data generation or augmentation:** Lin et al. [131] proposed multiple-layer feature-matching generative adversarial networks (MARTA GANs) for remote sensing data augmentation. MARTA GAN is based on DCGAN [5] however while DCGAN could produce images with a 64 × 64 resolution, MARTA GAN can produce 256×256 remote sensing images. To generate high-quality samples of remote images perceptual loss and feature-matching loss were used for model training. Mohandoss et al. [132] presented the MSG-ProGAN framework that uses ten bands of Sentinel-2 satellite imagery with varying resolutions to generate realistic multispectral imagery for data augmentation. To help with training stability the authors based their model on the MSGGAN [133], ProGAN [4] models and used WGAN-GP [20] loss function. Thus, MSG-ProGAN can generate multispectral 256 × 256 satellite images instead of RGB images.
- **Super Resolution:** HRPGAN [134] uses a PatchGAN inspired architecture to convert low resolution remote sensing images to high resolution images. The authors did not use batch normalization to preserve textures and sharp edges of ground objects in remote sensing images. Also, ReLU activations were replaced with SELU activations for overall lower training loss and stable training. In addition, the authors used a new loss function consisting of the traditional adversarial loss, perceptual reconstruction loss and regularization loss to train their model. D-SRGAN [135] converts low resolution Digital Elevation Models (DEMs) to high-resolution DEMs. D-SRGAN is based on the SRGAN [8] model. For training, D-SRGAN uses the combination of adversarial loss and content loss.

TABLE 6. Applications in biology.

Applications in Biology	GAN models
Protein Engineering	ProteinGAN [106], Antibody-GAN [107], FBGAN [108], DCGANs [109]
Data augmentation and data imputation	cscGAN [111],GGAN [112], scIGAN [114]
GANs for Biological Image Synthesis	CytoGAN [115], SGAN [117], DCGAN+Wasserstein loss [118]
Binding affinity prediction	GANsDTA [119]

TABLE 7. Applications in astronomy.

Applications in Astronomy	GAN models
Image to image translation	RadioGAN [120],pix2pix [6],pix2pixHD [24]
Image data generation and augmentation	SGAN [125], DCGAN [5], ProGAN [4], ExoGAN [128]
Image denoising	Spectra-GAN [130]

- **Pan-Sharpening:** Liu et al. [136] proposed PSGAN for solving the task of image pan-sharpening and carried out several experiments using different image datasets and different Generator architectures. PSGAN is superior to many popular pan-sharpening approaches in terms of generating high-quality pan-sharpened images with fine spatial details and high-fidelity spectral information under both low-scale and full-scale image settings, according to their experiments. Furthermore, the authors discovered that two-stream architecture is usually preferable to stacking and that the batch normalisation layer and the self-attention module are undesirable in pan-sharpening. Pan-GAN [137] uses one Generator and two Discriminators for performing pan sharpening. The Generator is based on the PNN [102] architecture but the image scale in the Generator remains the same in different layers. The spectral and spatial Discriminators are similar in structure but have different inputs. The generated HRMS image or the interpolated LRMS image is fed into the spectral Discriminator. The original panchromatic image or the single channel image generated by the generated HRMS image after average pooling along the channel dimension are the inputs for the spatial Discriminator.
- **Haze removal and Restoration:** Edge-sharpening cycle-consistent adversarial network (ES-CCGAN) [138] is a GAN-based unsupervised remote sensing image dehazing method based on the CycleGAN [7]. The authors used the unpaired image-to-image translation techniques for performing image dehazing. ES-CCGAN includes two generator networks and two discriminant networks. The Generators use DenseNet [139] blocks instead of the ResNet [140] block to generate dehazed remote-sensing images with plenty of texture information. An edge-sharpening loss was designed to restore clear edges in the images in addition to the adversarial loss, cycle-consistency loss and cyclic perceptual-consistency loss. Furthermore, to preserve contour information, a VGG16 [141] network was re-trained using remote-sensing image data to evaluate the perceptual loss. To tackle the problem of lack of availability of pairs of clear images and corresponding haze images to train the model, Sun et al. [142] proposed a cascade method combining two GANs. A learning-to-haze GAN(UGAN) learns to haze remote sensing images using unpaired clear and haze image sets. The UGAN then guides the learning-to-dehaze GAN (PAGAN) to learn how to dehaze UGAN hazed images. Wang et al. [143] developed the Image Despeckling Generative Adversarial Network (ID-GAN) to restore speckled Synthetic Aperture Radar (SAR) images. Their proposed method uses an encoder-decoder type architecture for the Generator which performs image despeckling by taking a noisy image as input. The Discriminator follows a standard layout with a sequence of convolution, batch normalization and ReLU layers, sigmoid function to distinguish between real and synthetic images. The authors used a refined loss function which is made up of pixel-to-pixel Euclidean loss, perceptual loss, and adversarial loss, all combined with appropriate weights.
- **Cloud Removal:** Several authors have used GANs for the removal of clouds contamination from remote sensing images([144], [145], [146], [147]). CLOUD-GAN [144] can translate cloudy images into cloud-free visible range images. CLOUD-GAN functions similar to CycleGAN [7] having two Generators and two Discriminators. The authors use the LSGAN [116] training method as it has been shown to generate higher quality images with a much more stable learning process compared to regular GANs. For thin cloud removal in multi-spectral images, Li et al. [145] proposed a novel semi-supervised method called CR-GAN-PM, which combines Generative Adversarial Networks and a physical model of cloud distortion. There are three networks in the CR-GAN-PM: an extraction network, a removal network, and a discriminative network. The GAN architecture is made up of the removal and discriminative networks. A combination of adversarial loss, reconstruction loss, correlation loss and optimization loss was used for training CR-GAN-PM.

G. MATERIAL SCIENCE

GANs have a wide range of applications in material science. GANs can be used to handle a variety of material science challenges such as Micro and crystal structure generation

TABLE 8. Applications in remote sensing.

Applications in Remote Sensing	GAN models
Data generation or augmentation	MARTA GAN [131],MSG-ProGAN [132]
Super Resolution	HRPGAN [134],D-SRGAN [135]
Pan-Sharpening	PSGAN [136],PAN-GAN [137]
Haze removal and Restoration	ES-CCGAN [138],ID-GAN [143], Sun et al. [142]
Cloud removal	CLOUD-GAN [144],CR-GAN-PM [145]

and design, Designing of complex architectured materials, Inorganic materials design, Virtual microstructure design and Topological design of metaporous materials for sound absorption.

Singh et al. [148] developed physics aware GAN model for the synthesis of binary microstructure images. The authors used three models to accomplish the task. The first model is the WGAN-GP [20]. The second approach replaces the usual Discriminator in a GAN with an invariance checker, which explicitly enforces known physical invariances. The third model combines the first two to recreate microstructures that adhere to both explicit physics invariances and implicit restrictions derived from image data. Yang et al. [149] proposed a GAN-based framework for microstructural materials design. A Bayesian optimization framework is used to obtain the microstructure with desired material property by processing the GAN generated latent variables. CrystalGAN [150] is a novel GAN-based framework for generating chemically stable crystallographic structures with enhanced domain complexity. The CrystalGAN model consists of three main components, a First step GAN, a Feature transfer procedure and the Second step GAN synthesizes. The First step GAN resembles the cross-domain GAN and generates pseudo-binary samples where the domains are mixed. The Feature transfer technique brings greater order complexity to the data generated from the samples obtained in the preceding stage. Finally, the second step GAN synthesizes ternary stable chemical compounds while adhering to geometric limitations. Kim et al. [151] proposed leveraging a coordinate-based crystal representation inspired by point clouds to generate crystal structures using generative adversarial network. Their Composition-Conditioned Crystal GAN can generate materials with the desired chemical composition by conditioning the network with a one-hot encoded composition vector.

Designing complex architectured materials is challenging and is heavily influenced by the experienced designers' prior knowledge. To tackle this issue, Mao et al. [152] successfully used GANs for the design of complex architectured materials. Millions of randomly generated architectured materials classified into different crystallographic symmetries were used to train their model. Their proposed model generates complex architectured designs that require no prior knowledge and can be readily applied in a wide range of applications.

Dan et al. proposed MatGAN [153] is the first GAN model for efficient sampling of inorganic materials design space by generating hypothetical inorganic materials. MatGAN, based on WGAN [19] can learn implicit chemical compositional rules from existing materials, allowing them to generate hypothetical yet chemically sound molecules. Another similar work was carried out by Hu et al. [154] where they used WGAN [19] to generate hypothetical inorganic materials consistent with the atomic combination of the training materials.

Lee et al. [155] employed DCGAN [5], CycleGAN [7] and Pix2Pix [6] to generate realistic virtual microstructural graph images. KL-divergence, a similarity metric that is considerably below 0.1, confirmed the similarity between the GAN-generated and ground truth images.

GANs were used by Zhang et al. [156] to greatly accelerate and improve the topological design of metaporous materials for sound absorption. Finite Element Method (FEM) simulation image data were used to train the model. The quality of the GAN generated designs was confirmed by FEM simulation and experimental evaluation, demonstrating that GANs are capable of generating metaporous material designs with satisfactory broadband absorption performance.

H. FINANCE

Financial data modeling is a challenging problem as there are complex statistical properties and dynamic stochastic factors behind the process. Many financial data are time-series data, such as real property price and stock market index. Many of them are very expensive to available and usually do not have enough labeled historical data, which greatly limits the performance of deep neural networks. In addition, unlike the static features, such as gender and image data, time-series data has a high temporal correlation across time. This becomes more complicated when we model multivariate time series where we need to consider the potentially complex dynamics of these variables across time. Recently, with the development and wide usage of GAN in image and audio tasks, a lot of research works have proposed to generate realistic time-series synthetic data in finance.

Efimov et al. [157] combine conditional GAN (CGAN) and Deep Regret Analytic Generate Adversarial Networks (DRAGANs) to replicate three American Express datasets with high fidelity. A regularization term is added in the Discriminator loss in DRAGANs to avoid gradient exploding or gradient vanishing effects as well as to stabilize the convergence. Zhou et al. [158] adopt the GAN-FD model (A GAN model for minimizing both forecast error loss and direction prediction loss) to predict stock prices. The Generator is based on LSTM layers while the Discriminator is using CNN layers. Li et al. [159] propose a conditional Wasserstin GAN (WGAN) named Stock-GAN to capture history dependency

TABLE 9. Applications in material science.

Applications in Material Science	GAN models
Micro and crystal structure generation and design	Hybrid (WGAN-GP [20]+GIN) [148], GAN+GP-Hedge Bayesian optimization framework [149], CrystalGAN [150], Composition-Conditioned Crystal GAN [151]
Designing complex architectured materials	GAN-based model [152]
Inorganic materials design	MatGAN [153], WGAN [19] based model [154]
Virtual microstructure design	(DCGAN [5], CycleGAN [7] and Pix2Pix [6]) [155]
Topological design of metaporous materials for sound absorption	GAN based model [156]

for stock market order streams. The proposed Generator network has two crafted features (1) approximating the double auction mechanism underlying stock exchanges and (2) including the order-book features as the condition information. Wiese et al. [39] introduce Quant GANs which use the Temporal Convolutional Networks (TCNs) architecture, also known as WaveNet [160] as the Generator. It shows the capability of capturing the long-range dependence such as the presence of volatility clusters in stock data such as S&P 500 index. FIN-GAN [161] captures the temporal structures of financial time-series so as to generate the major stylized facts of price returns, including the linear unpredictability, the fat-tailed distribution, volatility clustering, the leverage effects, the coarse-fine volatility correlation, and the gain/loss asymmetry. Leangarun et al. [162] build LSTM-GANs to detect the abnormal trading behaviors caused by stock price manipulations. The base architecture for both Generator and Discriminator is LSTM. The simulated manipulation cases are used for testing purposes. The detection system was tested with the trading data from the Stock Exchange of Thailand (SET) which achieves 68.1% accuracy in detecting pump-and-dump manipulations in unseen market data.

I. MARKETING

GANs can be leveraged to help businesses create effective marketing tools by synthesizing novel and unique designs for logos and generate fake images of models.

Typically designing a new logo is a fairly long and exhausting process and requires a lot of time and effort of the designer to meet the specific requirements of the clients. Sage et al. [163] put forward iWGAN, a GAN-based framework for virtually generating infinitely many variations of logos by specifying parameters such as shape, colour, and so on, in order to facilitate and expedite the logo design process. The authors proposed clustered GAN model to train on multi-modal data. Clustering was used to stabilise GAN training, prevent mode collapse and achieve higher quality samples on unlabeled datasets. The GAN models were based on the DCGAN [5] and WGAN-GP [20] models. LoGAN [164] or the Auxiliary Classifier Wasserstein Generative Adversarial Neural Network with gradient penalty (AC-WGAN-GP) is based on the ACGAN [97] architecture. LoGAN can be used to generate logos conditioned on twelve predetermined colors. LoGAN consists of a Generator, a Discriminator and a classification network to help the Discriminator in classifying the logos. The authors use the WGAN-GP [20] loss function for better training stability instead of using the ACGAN loss.

GANs can be used to replace real images of people for marketing-related ads, by generating synthetic images and videos thus alleviate problems related to privacy. Ma et al. [165] proposed Pose Guided Person Image Generation Network(PG^2) to generate synthetic fake images of a person in arbitrary poses conditioned on an image of a person and a new pose. PG^2 uses a two-stage process: Stage 1 Pose integration and Stage 2 Image refinement. Stage 1 generates a coarse output based on the input image and the target pose that depicts the human's overall structure. Stage 2 adopts the DCGAN [5] model and refine the initial result through adversarial training, resulting in sharper images. Deformable GANs [166] generate person images based on their appearance and pose. The authors introduced deformable skip connections and nearest neighbor loss to address large spatial deformation and to fix misalignment between the generated and ground-truth images. Song et al. [167] proposed E2E which uses GANs for unsupervised pose-guided person image generation. The authors break down the difficult task of learning a direct mapping under various poses into semantic parsing transformation and appearance generation to deal with its complexity.

J. FASHION DESIGNING

Fashion design is not the first thing that comes to mind when we think of GANs, however developing designs for clothes and outfits is another area where GANs have been utilized ([168], [169], [170]).

Based on the cGAN [18], Attribute-GAN [169] learns a mapping from a pair of outfits based on clothing attributes. The model has one Generator and two Discriminators. The Generator uses a U-Net architecture. A PatchGAN [171] Discriminator is used to capture local high-frequency structural information and a multi-task attribute classifier Discriminator is used to determine whether the generated fake apparel image has the expected ground truth attributes. Yuan et al. [170] presented Design-AttGAN, a new version of attribute GAN (AttGAN) [172] to edit garment images automatically based on certain user-defined attributes. The AttGAN's original formulation is changed to avoid the inherent conflict between the attribute classification loss and the reconstruction loss.

K. SPORTS

GANs can be used to generate sports texts, augment sports data, and predict and simulate sports activity to overcome the lack of labeled data and get insights into player behaviour patterns.

TABLE 10. Applications in finance.

Applications in Finance	GAN models
Financial Data Generation	CGAN and DRAGANs [157], FIN-GAN [161], Stock-GAN [159]
Stock Market Prediction	GAN-FD [158], Quant GANs [39]
Anomaly Detection in Finance	LSTM-GANs [162]

TABLE 11. Applications in marketing.

Applications in Marketing	GAN models
Logo generation	iWGAN [163], LoGAN [164]
Model generation and pose generation	PG^2 [165], Deformable GANs [166], E2E [167]

TABLE 12. Applications in fashion design.

Applications in Fashion Design	GAN models
Clothes and garment design	P-GANs [168],Attribute-GAN [169],Design-AttGAN [170]

Li et al. [173] used WGAN-GP [20] for automatic generation of sport news based on game stats. Their WGAN model takes scores as input and outputs sentences describing how one team defeated another. This paper also showed the potential applications of GANs in the NLP area.

JointsGAN [174] was proposed by Li et al. to augment soccer videos with a dribbling actions dataset for improving the accuracy of the dribbling styles classification model. The authors use Mask R-CNN [175] and OpenPose [176] to build dribbling player's joint model to act as a condition and guide the GAN model. The accuracy of classification is improved from 88.14% percent to 89.83% percent using the dribbling player's joints model as the condition to the GAN. Theagarajan et al. [177] used GANs to augment their dataset to build robust deep learning classification, object detection and localization models for soccer-related tasks. Specifically, they proposed the Triplet CNN-DCGAN framework to add more variability to the training set in order to improve the generalisation capacity of the aforementioned models. The GAN-based model is made of a regularizer CNN (i.e., Triplet CNN) along with DCGAN [5] and uses the DCGAN loss and the binary cross-entropy loss of the Triplet CNN.

Memory augmented Semi-Supervised Generative Adversarial Network (MSS-GAN) [178] can be used to predict the shot type and location in tennis. MSS-GAN is inspired by SS-GAN [179], coupled with memory modules to enhance its capabilities. The Perception Network (PN) is used to convert input images into embeddings, which are then combined with embeddings from the Episodic Memory (EM) and Semantic Memory (SM) to predict the next shot via the Response Generation Network (RGN). Finally, a GAN framework is used to train the network, with the RGN's predicted shot being passed to the Discriminator, which determines whether or not it is a realistic shot. BasketballGAN [180] is a cGAN [18] and WGAN [19] based framework to generate basketball set plays based on an input condition(offensive tactic sketched by coaches) and a random noise vector. The network was trained by minimizing the adversarial loss (Wasserstein loss [19]) dribbler loss, defender loss, ball passing loss, and acceleration loss.

L. MUSIC

Because human perception is sensitive to both global structure and fine-scale waveform coherence, music or audio synthesis is an intrinsically tough deep learning problem. As a result, synthesising music requires the creation of coherent raw audio waveforms that preserve both global and local structures. GANs have been applied for tasks such as music genre fusion and music generation.

FusionGAN [181] is a GAN based framework for unsupervised music genre fusion. The authors proposed the use of a multi-way GAN based model and utilized the Wasserstein distance measure as the objective function for stable training.

MidiNet [182] is a CNN-GAN based model to generate music with multiple MIDI channels. The model uses *conditioner* CNN to model the temporal dependencies by using the previous bar to condition the generation of the present bar, providing a powerful alternative to RNNs. The model features a flexible design that allows it to generate many genres of music based on input and specifications. Dong et al. [183] proposed Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment or MuseGAN. Based on GANs, MuseGAN can be used for symbolic multi-track music generation. Dong et al. [184] demonstrated a unique GAN-based model for producing binary-valued piano-rolls by employing binary neurons [185] as a refiner network in the Generator's output layer. When compared to existing approaches, the generated outputs of their model with deterministic binary neurons have fewer excessively fragmented notes. GANSYNTH [186], based on GANs can generate high-fidelity and locally-coherent audio. The proposed model outperforms the state-of-the-art WaveNet [160] model in generating high fidelity audio while also being much faster in sample generation. GANs were employed by Tokui [187] to create realistic rhythm patterns in unknown styles, which do not belong to any of the well-known electronic dance music genres in training data. Their proposed Creative-GAN model uses the Genre Ambiguity Loss to tackle the problem of originality. Li et al. [188] presented an inception model-based conditional generative adversarial network approach (INCO-GAN), which allows for the automatic synthesis of complete variable-length music. Their proposed model comprises of four distinct components: a conditional vector Generator (CVG), an inception model-based conditional GAN (INCO-GAN), a time distribution layer and an inception model [30]. Their analysis revealed that the proposed method's music is remarkably comparable to that created by human composers,

TABLE 13. Applications in sports.

Applications in Sports	GAN models
Sports text generation(NLP task)	WGAN-GP [20] based GAN [173]
Sports data augmentation	JointsGAN [174], Triplet CNN-DCGAN [177]
Sports action prediction and simulation	MSS-GAN [178], BasketballGAN [180]

with a cosine similarity of up to 0.987 between the frequency vectors. Muhamed et al. [189] presented the Transformer-GANs model, which combines GANs with Transformers to generate long, high-quality coherent music sequences. A pretrained SpanBERT [190] is used as the Discriminator and Transformer-XL [191] as the Generator. To train on long sequences, the authors use the Gumbel-Softmax technique [192] to obtain a differentiable approximation of the sampling process and to keep memory needs reasonable, a variant of the Truncated Backpropagation Through Time (TBPTT) algorithm [193] was utilised for gradient propagation over long sequences.

V. LIMITATIONS OF GANs AND FUTURE DIRECTION

In this section, we'll go through some of the issues that GANs encounter, notably those related to training stability. We also discuss some of the prospective research areas in which GAN productivity could be enhanced.

A. LIMITATIONS OF GANs

Generative adversarial networks (GANs) have gotten a lot of interest because of their capacity to use a lot of unlabeled data. While great progress has been achieved in alleviating some of the hurdles associated with developing and training novel GANs, there are still a few obstacles to overcome. We explain some of the most typical obstacles in training GANs, as well as some proposed strategies that attempt to mitigate such issues to some extent.

1) **Mode Collapse:** In most cases, we want the GAN to generate a wide range of outputs. For example, while creating photos of human faces we'd like the Generator to generate varied-looking faces with different features for every random input to Generator. Mode collapse happens when the Generator can produce only a single type of output or a small set of outputs. This may occur as a result of the Generator's constant search for the one output that appears most convincing to the Discriminator in order to easily trick the Discriminator, and hence continues to generate that one type.
   Several approaches have been proposed to alleviate the problem of mode collapse. Arjovsky et al. [19] found that Jensen-Shannon divergence is not ideal for measuring the distance of the distribution of the disjoint parts. As a result, they proposed the use of Wasserstein distance to calculate the distance between the produced and real data distributions. Metz et al. [194] proposed Unrolled Generative Adversarial Networks, which limit the risk of the Generator being over-optimized for a certain Discriminator, resulting in less mode collapse and increased stability.
2) **Non-convergence:** Although GANs are capable of achieving Nash equilibrium [1], arriving at this equilibrium is not straightforward. The training procedure necessitates maintaining balance and synchronisation between the Generator and Discriminator networks for optimal performance. Furthermore, only in the case of a convex function can gradient descent guarantee Nash equilibrium.
   Adding noise to Discriminator inputs and penalising Discriminator weights ([195], [196]) are two techniques of regularisation that authors have sought to utilise to improve GAN convergence.
3) **Vanishing Gradients:** Generator training can fail owing to vanishing gradients if the Discriminator is too good. A very accurate Discriminator produces gradients around zero, providing little feedback to the Generator and slowing or stopping learning.
   Goodfellow et al. [1] proposed a tweak to minimax loss to prevent the vanishing gradients problem. Although this tweak to the loss alleviates the vanishing gradients problem, it does not totally fix the problem, resulting in more unstable and oscillating training. The Wasserstein loss [19] is another technique to avoid vanishing gradients because it is designed to prevent vanishing gradients even when the Discriminator is trained to optimality.

B. FUTURE DIRECTION

Even though GANs have some limits and training issues, we simply cannot overlook their enormous potential as generative models. The most important area for future research is to make advancements in theoretical aspects to address concerns like mode collapse, vanishing gradients, non-convergence, and model breakdown. Changing learning objectives, regularising objectives, training procedures, tweaking hyperparameters, and other techniques have been proposed to overcome the aforementioned problems, as outlined in section V-A. In most cases, however, accomplishing these objectives entails a trade-off between desired output and training stability. As a result, rather than addressing one training issue at a time, future research in this field should use a holistic approach in order to achieve a breakthrough in theory to overcome the challenges mentioned above.

Besides overcoming the above theoretical aspects during model training, Saxena et al. [197] highlight some promising future research directions. (1) Keep high image quality without losing diversity. (2) Provide more theoretical analysis to better understand the tractable formulations during training and make training more stable and straightforward. (3) Improve the algorithm to make training efficient

TABLE 14. Applications in music.

Applications in Music	GAN models
Music genre fusion	FusionGAN [181]
Music generation	MidiNet [182], MuseGAN [183], Binary Neurons based WGAN-GP [184], GANSYNTH [186], Creative-GAN [187], INCO-GAN [188], Transformer-GANs [189]

(4) Combine other techniques, such as online learning, game theory, etc with GAN.

VI. CONCLUSION

In this paper, we presented state-of-the-art GAN models and their applications in a wide variety of domains. GANs' popularity stems from their ability to learn extremely nonlinear correlations between latent space and data space. As a result, the large amounts of unlabeled data that remain closed to supervised learning can be used. We discuss numerous aspects of GANs in this article, including theory, applications, and open research topics. We believe that this study will assist academic and industry researchers from various disciplines in gaining a full grasp of GANs and their possible applications. As a result, they will be able to analyse the potential application of GANs for their specific tasks.

ACKNOWLEDGMENT

The authors gratefully acknowledge this research is partially supported by Federal Highway Administration Exploratory Advanced Research 693JJ320C000021.

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, 2014, pp. 2672–2680, pp. 1–9.

[2] H. Gm, M. K. Gourisaria, M. Pandey, and S. S. Rautaray, "A comprehensive survey and analysis of generative models in machine learning," *Comput. Sci. Rev.*, vol. 38, Nov. 2020, Art. no. 100285.

[3] G. Hinton, *Boltzmann Machines*. Boston, MA, USA: Springer, 2010, pp. 132–136.

[4] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2017, *arXiv:1710.10196*.

[5] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks, 2016, *arXiv:1511.06434*.

[6] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976, doi: 10.1109/CVPR.2017.632.

[7] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2242–2251.

[8] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4681–4690.

[9] Y. Jiang and J. Li, "Generative adversarial network for image super-resolution combining texture loss," *Appl. Sci.*, vol. 10, no. 5, p. 1729, Mar. 2020.

[10] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8107–8116.

[11] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Computer Vision—ECCV*, L. Leal-Taixé and S. Roth, Eds. Cham, Switzerland: Springer, 2019, pp. 63–79.

[12] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5505–5514.

[13] L. Gonog and Y. Zhou, "A review: Generative adversarial networks," in *Proc. 14th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Jun. 2019, pp. 505–510.

[14] H. Alqahtani, M. Kavakli-Thorne, and G. Kumar, "Applications of generative adversarial networks (GANs): An updated review," *Arch. Comput. Methods Eng.*, vol. 28, no. 2, pp. 525–552, Mar. 2021.

[15] V. B. Raj and K. Hareesh, "Review on generative adversarial networks," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Jul. 2020, pp. 0479–0482.

[16] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: Algorithms, theory, and applications," 2020, *arXiv:2001.06937*.

[17] A. Aggarwal, M. Mittal, and G. Battineni, "Generative adversarial network: An overview of theory and applications," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 1, Apr. 2021, Art. no. 100004.

[18] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.

[19] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.

[20] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, 2017, pp. 1–11.

[21] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," 2016, *arXiv:1606.03657*.

[22] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5908–5916.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI* (Lecture Notes in Computer Science), vol. 9351, N. Navab, J. Hornegger, W. Wells, and A. Frangi, Eds. Cham, Switzerland: Springer, Oct. 2015, pp. 234–241.

[24] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8798–8807.

[25] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Jun. 2017, pp. 1510–1519.

[26] O. Mogren, "C-RNN-GAN: Continuous recurrent neural networks with adversarial training," 2016, *arXiv:1611.09904*.

[27] C. Esteban, S. L. Hyland, and G. Rätsch, "Real-valued (medical) time series generation with recurrent conditional GANs," 2017, *arXiv:1706.02633*.

[28] J. Yoon, D. Jarrett, and M. Van der Schaar, "Time-series generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–11.

[29] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, 2016, pp. 1–9.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[32] J. M. Joyce, *Kullback–Leibler Divergence*. Berlin, Germany: Springer, 2011, pp. 720–722.

[33] S. Gurumurthy, R. K. Sarvadevabhatla, and R. V. Babu, "DeLiGAN: Generative adversarial networks for diverse and limited data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4941–4949.

[34] T. Che, Y. Li, A. Paul Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," 2016, *arXiv:1612.02136*.

[35] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2017, pp. 6629–6640.

[36] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[37] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, 2003, pp. 1398–1402.

[38] A. Gretton, K. Borgwardt, M. J. Rasch, B. Scholkopf, and A. J. Smola, "A kernel method for the two-sample problem," 2008, *arXiv:0805.2368*.

[39] M. Wiese, R. Knobloch, R. Korn, and P. Kretschmer, "Quant GANs: Deep generation of financial time series," *Quant. Finance*, vol. 20, no. 9, pp. 1419–1440, Sep. 2020.

[40] C. Villani, *Optimal Transport: Old and New*, vol. 338. Berlin, Germany: Springer, 2009.

[41] W. Li, K. Zhou, L. Qi, L. Lu, N. Jiang, J. Lu, and J. Jia, "Best-buddy GANs for highly detailed image super-resolution," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 1412–1420.

[42] X. Zhu, L. Zhang, L. Zhang, X. Liu, Y. Shen, and S. Zhao, "GAN-based image super-resolution with a novel quality loss," *Math. Problems Eng.*, vol. 2020, pp. 1–12, Feb. 2020.

[43] T. Vu, T. M. Luu, and C. D. Yoo, "Perception-enhanced image super-resolution via relativistic generative adversarial networks," in *Computer Vision—ECCV*, L. Leal-Taixe and Stefan Roth, Eds. Cham, Switzerland: Springer, 2019, pp. 98–113.

[44] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 11, pp. 3943–3956, Nov. 2020.

[45] G. Perarnau, J. van de Weijer, B. Raducanu, and J. M. Álvarez, "Invertible conditional GANs for image editing," 2016, *arXiv:1611.06355*.

[46] P. Zhuang, O. Koyejo, and A. G. Schwing, "Enjoy your editing: Controllable GANs for image editing via latent space navigation," 2021, *arXiv:2102.01187*.

[47] T. Zhang, W.-H. Tian, T.-Y. Zheng, Z.-N. Li, X.-M. Du, and F. Li, "Realistic face image generation based on generative adversarial network," in *Proc. 16th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process.*, Dec. 2019, pp. 303–306.

[48] Z. Zhang, X. Pan, S. Jiang, and P. Zhao, "High-quality face image generation based on generative adversarial networks," *J. Vis. Commun. Image Represent.*, vol. 71, Aug. 2020, Art. no. 102719.

[49] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," 2018, *arXiv:1812.04948*.

[50] A. Bhattacharjee, S. Banerjee, and S. Das, "Posix-GAN: Generating multiple poses using GAN for pose-invariant face recognition," in *Computer Vision—ECCV*, L. Leal-Taixe and S. Roth, Eds. Cham, Switzerland: Springer, 2019, pp. 427–443.

[51] M. Chen, C. Li, K. Li, H. Zhang, and X. He, "Double encoder conditional GAN for facial expression synthesis," in *Proc. 37th Chin. Control Conf. (CCC)*, Jul. 2018, pp. 9286–9291.

[52] H. Tang, W. Wang, S. Wu, X. Chen, D. Xu, N. Sebe, and Y. Yan, "Expression conditional GAN for facial expression-to-expression translation," 2019, *arXiv:1905.05416*.

[53] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," 2017, *arXiv:1703.05192*.

[54] Y. Choi, M. Choi, M. Kim, J. W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," 2017, *arXiv:1711.09020*.

[55] T. Li, R. Qian, C. Dong, S. Liu, Q. Yan, W. Zhu, and L. Lin, "BeautyGAN: Instance-level facial makeup transfer with deep generative adversarial network," in *Proc. 26th ACM Int. Conf. Multimedia*, New York, NY, USA, Oct. 2018, pp. 645–653.

[56] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6882–6890.

[57] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2016, pp. 613–621.

[58] M. Saito, E. Matsumoto, and S. Saito, "Temporal generative adversarial nets with singular value clipping," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2830–2839.

[59] K. Ohnishi, S. Yamamoto, Y. Ushiku, and T. Harada, "Hierarchical video generation from orthogonal information: Optical flow and texture," 2017, *arXiv:1711.09618*.

[60] S. Tulyakov, M. Liu, X. Yang, and J. Kautz, "MoCoGAN: Decomposing motion and content for video generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1526–1535.

[61] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," 2019, *arXiv:1809.11096*.

[62] A. Clark, J. Donahue, and K. Simonyan, "Adversarial video generation on complex datasets," 2019, *arXiv:1907.06571*.

[63] Y. Li, M. R. Min, D. Shen, D. Carlson, and L. Carin, "Video generation from text," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1–8.

[64] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2014, *arXiv:1312.6114*.

[65] Y. Balaji, M. R. Min, B. Bai, R. Chellappa, and H. P. Graf, "Conditional GAN with discriminative filter generation for text-to-video synthesis," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1995–2001.

[66] Y. Li, Z. Gan, Y. Shen, J. Liu, Y. Cheng, Y. Wu, L. Carin, D. Carlson, and J. Gao, "StoryGAN: A sequential conditional GAN for story visualization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Los Alamitos, CA, USA, Jun. 2019, pp. 6322–6331.

[67] Q. Chen, Q. Wu, J. Chen, Q. Wu, A. van den Hengel, and M. Tan, "Scripted video generation with a bottom-up generative adversarial network," *IEEE Trans. Image Process.*, vol. 29, pp. 7454–7467, 2020.

[68] D. Kim, D. Joo, and J. Kim, "TiVGAN: Text to image to video generation with step-by-step evolutionary generator," *IEEE Access*, vol. 8, pp. 153113–153122, 2020.

[69] S. A. Jalalifar, H. Hasani, and H. Aghajan, "Speech-driven facial reenactment using conditional generative adversarial networks," 2018, *arXiv:1803.07461*.

[70] K. Vougioukas, S. Petridis, and M. Pantic, "End-to-end speech-driven realistic facial animation with temporal GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, Jun. 2019, pp. 37–40.

[71] H. Zhou, Y. Liu, Z. Liu, P. Luo, and X. Wang, "Talking face generation by adversarially disentangled audio-visual representation," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 9299–9306.

[72] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," 2016, *arXiv:1511.05440*.

[73] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, "Decomposing motion and content for natural video sequence prediction," 2018, *arXiv:1706.08033*.

[74] S. Aigner and M. Körner, "FutureGAN: Anticipating the future frames of video sequences using spatio-temporal 3D convolutions in progressively growing GANs," 2018, *arXiv:1810.01325*.

[75] Z. Hu and J. T. L. Wang, "A novel adversarial inference framework for video prediction with action control," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 768–772.

[76] X. Liang, L. Lee, W. Dai, and E. P. Xing, "Dual motion GAN for future-flow embedded video prediction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1762–1770.

[77] C. Vondrick and A. Torralba, "Generating the future with adversarial transformers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2992–3000.

[78] C. Lu, M. Hirsch, and B. Schölkopf, "Flexible spatio-temporal networks for video prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2137–2145.

[79] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2015, pp. 4489–4497.

[80] P. Bhattacharjee and S. Das, "Temporal coherency based criteria for predicting video frames using deep multi-stage generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, 2017, pp. 1–10.

[81] Y.-H. Kwon and M.-G. Park, "Predicting future frames using retrospective cycle GAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1811–1820.

[82] C.-B. Jin, H. Kim, M. Liu, W. Jung, S. Joo, E. Park, Y. S. Ahn, I. H. Han, J. I. Lee, and X. Cui, "Deep CT to MR synthesis using paired and unpaired data," *Sensors*, vol. 19, no. 10, p. 2361, 2019.

[83] Z. Yu, Q. Xiang, J. Meng, C. Kou, Q. Ren, and Y. Lu, "Retinal image synthesis from multiple-landmarks input with generative adversarial networks," *Biomed. Eng. OnLine*, vol. 18, no. 1, p. 62, May 2019.

[84] H. Zhao, H. Li, S. Maurer-Stroh, and L. Cheng, "Synthesizing retinal and neuronal images with generative adversarial nets," *Med. Image Anal.*, vol. 49, pp. 14–26, Oct. 2018.

[85] K. Armanious, C. Jiang, M. Fischer, T. Küstner, T. Hepp, K. Nikolaou, S. Gatidis, and B. Yang, "MedGAN: Medical image translation using GANs," *Computerized Med. Imag. Graph.*, vol. 79, Jan. 2020, Art. no. 101684.

[86] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*.

[87] V. Sandfort, K. Yan, P. J. Pickhardt, and R. M. Summers, "Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks," *Sci. Rep.*, vol. 9, no. 1, p. 16884, Nov. 2019.

[88] A. Gohorbani, V. Natarajan, D. D. Coz, and Y. Liu, "DermGAN: Synthetic generation of clinical skin images with pathology," 2019, *arXiv:1911.08716*.

[89] L. Middel, C. Palm, and M. Erdt, "Synthesis of medical images using GANs," in *Uncertainty for Safe Utilization of Machine Learning in Medical Imaging and Clinical Image-Based Procedures*, H. Greenspan, R. Tanno, M. Erdt, T. Arbel, C. Baumgartner, A. Dalca, C. H. Sudre, W. M. Wells, K. Drechsler, M. G. Linguraru, C. O. Laura, R. Shekhar, S. Wesarg, and M. A. G. Ballester, Eds. Cham, Switzerland: Springer, 2019, pp. 125–134.

[90] M. J. M. Chuquicusma, S. Hussein, J. Burt, and U. Bagci, "How to fool radiologists with generative adversarial networks? A visual Turing test for lung cancer diagnosis," in *Proc. IEEE 15th Int. Symp. Biomed. Imag.*, 2018, pp. 240–244.

[91] P. Costa, A. Galdran, M. I. Meyer, M. Niemeijer, M. Abramoff, A. M. Mendonça, and A. Campilho, "End-to-end adversarial retinal image synthesis," *IEEE Trans. Med. Imag.*, vol. 37, no. 3, pp. 781–791, Mar. 2018.

[92] Q. Zhang, H. Wang, H. Lu, D. Won, and S. W. Yoon, "Medical image synthesis with generative adversarial networks for tissue recognition," in *Proc. IEEE Int. Conf. Healthcare Informat. (ICHI)*, Los Alamitos, CA, USA, Jun. 2018, pp. 199–207.

[93] A. Beers, J. M. Brown, K. Chang, J. P. Campbell, S. Ostmo, M. F. Chiang, and J. Kalpathy-Cramer, "High-resolution medical image synthesis using progressively grown generative adversarial networks," 2018, *ArXiv:1805.03144*.

[94] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: Boundary equilibrium generative adversarial networks," 2017, *arXiv:1703.10717*.

[95] T. Nagasawa, T. Sato, I. Nambu, and Y. Wada, "FNIRS-GANs: Data augmentation using generative adversarial networks for classifying motor tasks from functional near-infrared spectroscopy," *J. Neural Eng.*, vol. 17, no. 1, Feb. 2020, Art. no. 016068.

[96] E. Piacentino, A. Guarner, and C. Angulo, "Generating synthetic ECGs using GANs for anonymizing healthcare data," *Electronics*, vol. 10, no. 4, p. 389, Feb. 2021.

[97] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2642–2651.

[98] C. Kwon, S. Park, S. Ko, and J. Ahn, "Increasing prediction accuracy of pathogenic staging by sample augmentation with a GAN," *PLoS ONE*, vol. 16, no. 4, Apr. 2021, Art. no. e0250458.

[99] Y. Chen, F. Shi, A. G. Christodoulou, Z. Zhou, Y. Xie, and D. Li, "Efficient and accurate MRI super-resolution using a generative adversarial network and 3D multi-level densely connected network," 2018, *arXiv:1803.01417*.

[100] Y. Gu, Z. Zeng, H. Chen, J. Wei, Y. Zhang, B. Chen, Y. Li, Y. Qin, Q. Xie, Z. Jiang, and Y. Lu, "MedSRGAN: Medical images super-resolution using generative adversarial networks," *Multimedia Tools Appl.*, vol. 79, no. 29, pp. 21815–21840, Aug. 2020.

[101] K. Yamashita and K. Markov, "Medical image enhancement using super resolution methods," in *Computational Science—ICCS*, V. V. Krzhizhanovskaya, G. Zavodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos, and J. Teixeira, Eds. Cham, Switzerland: Springer, 2020, pp. 496–508.

[102] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[103] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1646–1654.

[104] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1637–1645.

[105] J. G. Fujimoto, C. Pitris, S. A. Boppart, and M. E. Brezinski, "Optical coherence tomography: An emerging technology for biomedical imaging and optical biopsy," *Neoplasia*, vol. 2, nos. 1–2, pp. 9–25, Jan. 2000.

[106] D. Repecka, V. Jauniskis, L. Karpus, E. Rembeza, I. Rokaitis, J. Zrimec, S. Poviloniene, A. Laurynenas, S. Viknander, W. Abuajwa, O. Savolainen, R. Meskys, M. K. M. Engqvist, and A. Zelezniak, "Expanding functional protein sequence spaces using generative adversarial networks," *Nature Mach. Intell.*, vol. 3, no. 4, pp. 324–333, Apr. 2021.

[107] T. Amimeur, J. M. Shaver, R. R. Ketchem, J. A. Taylor, R. H. Clark, J. Smith, D. Van Citters, C. C. Siska, P. Smidt, M. Sprague, B. A. Kerwin, and D. Pettit, "Designing feature-controlled humanoid antibody discovery libraries using generative adversarial networks," *BioRxiv*, 2020.

[108] A. Gupta and J. Zou, "Feedback GAN for dna optimizes protein functions," *Nature Mach. Intell.*, vol. 1, no. 2, pp. 105–111, Feb. 2019.

[109] N. Anand and P. Huang, "Generative modeling for protein structures," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, 2018, pp. 1–12.

[110] A. Ghahramani, F. M. Watt, and N. M. Luscombe, "Generative adversarial networks simulate gene expression and predict perturbations in single cells," *BioRxiv*, 2018.

[111] M. Marouf, P. Machart, V. Bansal, C. Kilian, D. S. Magruder, C. F. Krebs, and S. Bonn, "Realistic in silico generation and augmentation of single-cell RNA-seq data using generative adversarial networks," *Nature Commun.*, vol. 11, no. 1, p. 166, Jan. 2020.

[112] X. Wang, K. G. Dizaji, and H. Huang, "Conditional generative adversarial network for gene expression inference," *Bioinformatics*, vol. 34, no. 17, pp. 603–611, Sep. 2018.

[113] J. Park, H. Kim, J. Kim, and M. Cheon, "A practical application of generative adversarial networks for RNA-seq analysis to predict the molecular progress of Alzheimer's disease," *PLOS Comput. Biol.*, vol. 16, no. 7, Jul. 2020, Art. no. e1008099.

[114] Y. Xu, Z. Zhang, L. You, J. Liu, Z. Fan, and X. Zhou, "ScIGANs: Single-cell RNA-seq imputation using generative adversarial networks," *Nucleic Acids Res.*, vol. 48, no. 15, p. e85, Sep. 2020.

[115] P. Goldsborough, N. Pawlowski, J. C. Caicedo, S. Singh, and A. E. Carpenter, "CytoGAN: Generative modeling of cell images," *BioRxiv*, 2017.

[116] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2813–2821.

[117] L. Han, R. F. Murphy, and D. Ramanan, "Learning generative models of tissue organization with supervised GANs," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 682–690.

[118] A. Osokin, A. Chessel, R. E. C. Salas, and F. Vaggi, "GANs for biological image synthesis," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2252–2261.

[119] L. Zhao, J. Wang, L. Pang, Y. Liu, and J. Zhang, "GANsDTA: Predicting drug-target binding affinity using GANs," *Frontiers Genet.*, vol. 10, p. 1243, Jan. 2020.

[120] N. Glaser, O. I. Wong, K. Schawinski, and C. Zhang, "RadioGAN—Translations between different radio surveys with generative adversarial networks," *Monthly Notices Roy. Astronomical Soc.*, vol. 487, no. 3, pp. 4190–4207, Aug. 2019.

[121] E. Park, Y.-J. Moon, J.-Y. Lee, R.-S. Kim, H. Lee, D. Lim, G. Shin, and T. Kim, "Generation of solar UV and EUV images from SDO/HMI magnetograms by deep learning," *Astrophysical J. Lett.*, vol. 884, no. 1, p. L23, Oct. 2019.

[122] T. Kim, E. Park, H. Lee, Y.-J. Moon, S.-H. Bae, D. Lim, S. Jang, L. Kim, I.-H. Cho, M. Choi, and K.-S. Cho, "Solar farside magnetograms from deep learning analysis of STEREO/EUVI data," *Nature Astron.*, vol. 3, no. 5, pp. 397–400, May 2019.

[123] H.-J. Jeong, Y.-J. Moon, E. Park, and H. Lee, "Solar coronal magnetic field extrapolation from synchronic data with AI-generated farside," *Astrophysical J. Lett.*, vol. 903, no. 2, p. 25, Nov. 2020.

[124] G. Shin, Y.-J. Moon, E. Park, H.-J. Jeong, H. Lee, and S.-H. Bae, "Generation of high-resolution solar pseudo-magnetograms from Ca II K images by deep learning," *Astrophysical J. Lett.*, vol. 895, no. 1, p. 16, May 2020.

[125] M. J. Smith and J. E. Geach, "Generative deep fields: Arbitrarily sized, random synthetic astronomical images through deep learning," *Monthly Notices Roy. Astronomical Soc.*, vol. 490, no. 4, pp. 4985–4990, Dec. 2019.

[126] M. Ullmo, A. Decelle, and N. Aghanim, "Encoding large scale cosmological structure with generative adversarial networks," *Astron. Astrophys.*, vol. 651, p. A46, Jul. 2020.

[127] M. Dia, E. Savary, M. Melchior, and F. Courbin, "Galaxy image simulation using progressive GANs," 2019, *arXiv:1909.12160*.

[128] T. Zingales and I. P. Waldmann, "ExoGAN: Retrieving exoplanetary atmospheres using deep convolutional generative adversarial networks," *Astronomical J.*, vol. 156, no. 6, p. 268, Nov. 2018.

[129] L. Fussell and B. Moews, "Forging new worlds: High-resolution synthetic galaxies with chained generative adversarial networks," *Monthly Notices Roy. Astronomical Soc.*, vol. 485, no. 3, pp. 3203–3214, Jan. 2019.

[130] M. Wu, Y. Bu, J. Pan, Z. Yi, and X. Kong, "Spectra-GANs: A new automated denoising method for low-S/N stellar spectra," *IEEE Access*, vol. 8, pp. 107912–107926, 2020.

[131] D. Lin, K. Fu, Y. Wang, G. Xu, and X. Sun, "MARTA GANs: Unsupervised representation learning for remote sensing image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 2092–2096, Nov. 2017.

[132] T. Mohandoss, A. Kulkarni, D. Northrup, E. Mwebaze, and H. Alemohammad, "Generating synthetic multispectral satellite imagery from Sentinel-2," 2020, *arXiv:2012.03108*.

[133] A. Karnewar and O. Wang, "MSG-GAN: Multi-scale gradients for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 7799–7808.

[134] H. Sun, P. Wang, Y. Chang, L. Qi, H. Wang, D. Xiao, C. Zhong, X. Wu, W. Li, and B. Sun, "HRPGAN: A GAN-based model to generate high-resolution remote sensing images," *IOP Conf. Ser., Earth Environ. Sci.*, vol. 428, Jan. 2020, Art. no. 012060.

[135] B. Z. Demiray, M. Sit, and I. Demir, "D-SRGAN: DEM super-resolution with generative adversarial networks," *Social Netw. Comput. Sci.*, vol. 2, no. 1, p. 48, Jan. 2021.

[136] X. Liu, Y. Wang, and Q. Liu, "PSGAN: A generative adversarial network for remote sensing image pan-sharpening," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 873–877.

[137] J. Ma, W. Yu, C. Chen, P. Liang, X. Guo, and J. Jiang, "Pan-GAN: An unsupervised pan-sharpening method for remote sensing image fusion," *Inf. Fusion*, vol. 62, pp. 110–120, Oct. 2020.

[138] A. Hu, Z. Xie, Y. Xu, M. Xie, L. Wu, and Q. Qiu, "Unsupervised haze removal for high-resolution optical remote-sensing images based on improved generative adversarial networks," *Remote Sens.*, vol. 12, no. 24, p. 4162, Dec. 2020.

[139] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[140] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 770–778.

[141] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, *arXiv:1409.1556*.

[142] X. Sun and J. Xu, "Remote sensing images dehazing algorithm based on cascade generative adversarial networks," in *Proc. 13th Int. Congr. Image Signal Process., Biomed. Eng. Informat. (CISP-BMEI)*, 2020, pp. 316–321.

[143] P. Wang, H. Zhang, and V. M. Patel, "Generative adversarial network-based restoration of speckled SAR images," in *Proc. IEEE 7th Int. Workshop Comput. Adv. Multi-Sensor Adapt. Process. (CAMSAP)*, Dec. 2017, pp. 1–5.

[144] P. Singh and N. Komodakis, "Cloud-GAN: Cloud removal for Sentinel-2 imagery using a cyclic consistent generative adversarial networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2018, pp. 1772–1775.

[145] J. Li, Z. Wu, Z. Hu, J. Zhang, M. Li, L. Mo, and M. Molinier, "Thin cloud removal in optical remote sensing images based on generative adversarial networks and physical model of cloud distortion," *ISPRS J. Photogramm. Remote Sens.*, vol. 166, pp. 373–389, Aug. 2020.

[146] H. Pan, "Cloud removal for remote sensing imagery via spatial attention generative adversarial network," 2020, *arXiv:2009.13015*.

[147] X. Wen, Z. Pan, Y. Hu, and J. Liu, "Generative adversarial learning in YUV color space for thin cloud removal on satellite imagery," *Remote Sens.*, vol. 13, no. 6, p. 1079, Mar. 2021.

[148] R. Singh, V. Shah, B. Pokuri, S. Sarkar, B. Ganapathysubramanian, and C. Hegde, "Physics-aware deep generative models for creating synthetic microstructures," 2018, *arXiv:1811.09669*.

[149] Z. Yang, X. Li, L. Catherine Brinson, A. N. Choudhary, W. Chen, and A. Agrawal, "Microstructural materials design via deep adversarial learning methodology," *J. Mech. Des.*, vol. 140, no. 11, Nov. 2018, Art. no. 111416.

[150] A. Nouira, N. Sokolovska, and J.-C. Crivello, "CrystalGAN: Learning to discover crystallographic structures with generative adversarial networks," 2019, *arXiv:1810.11203*.

[151] S. Kim, J. Noh, G. H. Gu, A. Aspuru-Guzik, and Y. Jung, "Generative adversarial networks for crystal structure prediction," *ACS Central Sci.*, vol. 6, no. 8, pp. 1412–1420, 2020.

[152] Y. Mao, Q. He, and X. Zhao, "Designing complex architectured materials with generative adversarial networks," *Sci. Adv.*, vol. 6, Apr. 2020, Art. no. eaaz4169.

[153] Y. Dan, Y. Zhao, X. Li, S. Li, M. Hu, and J. Hu, "Generative adversarial networks (GAN) based efficient sampling of chemical composition space for inverse design of inorganic materials," *Npj Comput. Mater.*, vol. 6, no. 1, p. 84, Jun. 2020.

[154] T. Hu, H. Song, T. Jiang, and S. Li, "Learning representations of inorganic materials from generative adversarial networks," *Symmetry*, vol. 12, no. 11, p. 1889, Nov. 2020.

[155] J. Lee, N. H. Goo, W. B. Park, M. Pyo, and K. Sohn, "Virtual microstructure design for steels using generative adversarial networks," *Eng. Rep.*, vol. 3, no. 1, p. e12274, Jan. 2021.

[156] H. Zhang, Y. Wang, H. Zhao, K. Lu, D. Yu, and J. Wen, "Accelerated topological design of metaporous materials of broadband sound absorption performance by generative adversarial networks," *Mater. Des.*, vol. 207, Sep. 2021, Art. no. 109855.

[157] D. Efimov, D. Xu, L. Kong, A. Nefedov, and A. Anandakrishnan, "Using generative adversarial networks to synthesize artificial financial datasets," 2020, *arXiv:2002.02271*.

[158] X. Zhou, Z. Pan, G. Hu, S. Tang, and C. Zhao, "Stock market prediction on high-frequency data using generative adversarial nets," *Math. Problems Eng.*, vol. 2018, pp. 1–11, 2018.

[159] J. Li, X. Wang, Y. Lin, A. Sinha, and M. Wellman, "Generating realistic stock market order streams," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 727–734.

[160] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," 2016, *arXiv:1609.03499*.

[161] S. Takahashi, Y. Chen, and K. Tanaka-Ishii, "Modeling financial time-series with generative adversarial networks," *Phys. A, Stat. Mech. Appl.*, vol. 527, Aug. 2019, Art. no. 121261.

[162] T. Leangarun, P. Tangamchit, and S. Thajchayapong, "Stock price manipulation detection using generative adversarial networks," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2018, pp. 2104–2111.

[163] A. Sage, R. Timofte, E. Agustsson, and L. V. Gool, "Logo synthesis and manipulation with clustered generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5879–5888.

[164] A. Mino and G. Spanakis, "LoGAN: Generating logos with a generative adversarial neural network conditioned on color," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2018, pp. 965–970.

[165] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool, "Pose guided person image generation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, 2017, pp. 1–11.
[166] A. Siarohin, E. Sangineto, S. Lathuilière, and N. Sebe, "Deformable GANs for pose-based human image generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3408–3416.
[167] S. Song, W. Zhang, J. Liu, and T. Mei, "Unsupervised person image generation with semantic parsing transformation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2352–2361.
[168] N. Kato, H. Osone, K. Oomori, C. W. Ooi, and Y. Ochiai, "GANs-based clothes design: Pattern maker is all you need to design clothing," in *Proc. 10th Augmented Human Int. Conf.*, New York, NY, USA, Mar. 2019, pp. 1–7.
[169] L. Liu, H. Zhang, Y. Ji, and Q. M. Jonathan Wu, "Toward AI fashion design: An attribute-GAN model for clothing match," *Neurocomputing*, vol. 341, pp. 156–167, May 2019.
[170] C. Yuan and M. Moghaddam, "Garment design with generative adversarial networks," 2020, *arXiv:2007.10947*.
[171] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," 2016, *arXiv:1611.07004*.
[172] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, "AttGAN: Facial attribute editing by only changing what you want," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5464–5478, Nov. 2019.
[173] C. Li, Y. Su, J. Qi, and M. Xiao, "Using GAN to generate sport news from live game stats," in *Cognitive Computing—ICCC*, R. Xu, J. Wang, and L.-J. Zhang, Eds. Cham, Switzerland: Springer, 2019, pp. 102–116.
[174] R. Li and B. Bhanu, "Fine-grained visual dribbling style analysis for soccer videos with augmented dribble energy image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 2439–2447.
[175] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[176] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, Jan. 2021.
[177] R. Theagarajan and B. Bhanu, "An automated system for generating tactical performance statistics for individual soccer players from videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 2, pp. 632–646, Feb. 2021.
[178] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Memory augmented deep generative models for forecasting the next shot location in tennis," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 9, pp. 1785–1797, Sep. 2020.
[179] E. Denton, S. Gross, and R. Fergus, "Semi-supervised learning with context-conditional generative adversarial networks," 2016, *arXiv:1611.06430*.
[180] H.-Y. Hsieh, C.-Y. Chen, Y.-S. Wang, and J.-H. Chuang, "BasketballGAN: Generating basketball play simulation through sketching," in *Proc. 27th ACM Int. Conf. Multimedia*, New York, NY, USA, Oct. 2019, pp. 720–728.
[181] Z. Chen, C.-W. Wu, Y.-C. Lu, A. Lerch, and C.-T. Lu, "Learning to fuse music genres with generative adversarial dual learning," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2017, pp. 817–822.
[182] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation," in *Proc. ISMIR*, 2017, pp. 1–8.
[183] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1–8.
[184] H.-W. Dong and Y.-H. Yang, "Convolutional generative adversarial networks with binary neurons for polyphonic music generation," in *Proc. ISMIR*, 2018, pp. 1–13.
[185] R2RT, *Binary Stochastic Neurons in TensorFlow*.
[186] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "GANSynth: Adversarial neural audio synthesis," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–17.
[187] N. Tokui, "Can GAN originate new electronic dance music genres? Generating novel rhythm patterns using GAN with genre ambiguity loss," 2020, *arXiv:2011.13062*.
[188] S. Li and Y. Sung, "INCO-GAN: Variable-length music generation method based on inception model-based conditional GAN," *Mathematics*, vol. 9, no. 4, p. 387, 2021.
[189] A. Muhamed, L. Li, X. Shi, S. Yaddanapudi, W. Chi, D. Jackson, R. Suresh, Z. C. Lipton, and A. Smola, "Symbolic music generation with transformer-GANs," in *Proc. AAAI*, 2021, pp. 408–417.
[190] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "SpanBERT: Improving pre-training by representing and predicting spans," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 64–77, Dec. 2020.
[191] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proc. ACL*, 2019, pp. 1–20.
[192] M. J. Kusner and J. M. Hernández-Lobato, "GANS for sequences of discrete elements with the gumbel-softmax distribution," 2016, *arXiv:1611.04051*.
[193] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. NIPS*, 2014, pp. 1–9.
[194] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," 2017, *arXiv:1611.02163*.
[195] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," 2017, *arXiv:1701.04862*.
[196] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann, "Stabilizing training of generative adversarial networks through regularization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.
[197] D. Saxena and J. Cao, "Generative adversarial networks (GANs): Challenges, solutions, and future directions," *ACM Comput. Surv.*, vol. 54, no. 3, pp. 1–42, Apr. 2022.

ANKAN DASH received the master's degree in aerospace engineering from the University of Illinois at Urbana–Champaign. He is currently pursuing the Ph.D. degree in computer science with the New Jersey Institute of Technology. His areas of research focus on computer vision, 3-D computer vision, and deep learning.

JUNYI YE received the first master's degree in optics from Shanghai University, in 2015, and the second master's degree in computer science from the New Jersey Institute of Technology, in 2019. He is currently pursuing the Ph.D. degree with the Ying Wu College of Computing, New Jersey Institute of Technology. His research interests include time series prediction for the stock market, time delay estimation, and computer vision.

GUILING WANG received the Ph.D. degree in computer science and engineering and a minor in statistics from The Pennsylvania State University, in 2006. She is currently a Distinguished Professor and the Associate Dean of research with the Ying Wu College of Computing, New Jersey Institute of Technology. She also holds a joint appointment with the Data Science Department, Martin Tuchman School of Management. Her research interests include FinTech, applied deep learning, blockchain technologies, and intelligent transportation.