Generative Adversarial Networks: A Comprehensive Review and the Way Forward

Boriane Y. Tchaleu, Alain R. Ndjiongue, *Senior Member*, *IEEE*, and Collins A. Leke

Abstract—**The deep learning ability to recognize patterns in data has recently become popular within education. Created in 2014, generative adversarial networks (GANs) are innovative classes of deep learning generative models based on game theory and consist of two players. GANs generate data from scratch using two neural networks: the generator and the discriminator. Since their creation, GANs have been utilized in many applications and have advantages and disadvantages. In light of such a long journey, evaluating the technology is essential as it provides readers with the way forward. To this end, this paper reviews GANs and explores some fundamental challenges that develop during evaluation and training. We also discuss GANs' challenges and elaborate subsequent solutions. Through a single context, we explain the reasoning behind the GAN technology and examine its direction and motivation. We discuss different variants of GANs and real-world application examples, including performance evaluation metrics across various sectors. We consider results obtained recently and highlight ideas for further investigation. This detailed retrospect will give the reader a better understanding of the possible uses of GANs. It will also show how they can help address current issues in a variety of disciplines. Before that, the paper reviews GANs' architectures and network approaches and elaborates on challenges and solutions. The reader is then guided through the literature on the various applications of GANs and the importance of the research interest associated with GANs. As a final step, we suggest the way forward and conclude the review.**

Index Terms—**Generative adversarial network (GAN), artificial intelligence, generator, discriminator, deep fake, mode collapse, neural network.**

List of Abbreviations

ACGAN	Auxiliary classifier generative adversarial network
AI	Artificial intelligence
BigGAN	Big generative adversarial network
BRISQUE	Blind image spatial quality evaluator
cGAN	Conditional generative adversarial network
CT	Computed tomography
CycleGAN	Cycle generative adversarial network
DCGAN	Deep convolutional generative adversarial network
DGM	Deep generative model
D	Discriminator
DS	Diversity score
DVD-GAN	Dual video discriminator generative adversarial network
EBGAN	Energy-based generative adversarial network
EM	Earth movers distance
ERGAS	Error relative global dimensionless synthesis
FID	Frecher inception distance
FSGAN	Face swapping generative adversarial network
GAN	Generative adversarial network
G	Generator
InfoGAN	Information maximizing generative adversarial network
IS	Inspection score
KID	Kernel inception distance
KL	Kullback-Leibler
LAPGAN	Laplacian pyramid generative adversarial network
LPIPS	Learned perceptual image patch similarity
LSGAN	Least square generative adversarial network
MIS	Modified inception score
ML	Machine learning
MMD	Maximum mean discrepancy
MNIST	Modified National Institute of Standards and Technology Database
MRGAN	Mode regularized generative adversarial network
MRI	Magnetic resonance imaging
MS	Mode score
NAS	Nash equilibrium
Pix2Pix GAN	Pixel-to-pixel generative adversarial network
PGAN	Progressive generative adversarial network
PSNR	Peak signal-to-noise ratio
PPL	Perceptual path length
PR	Precision and recall
RMSE	Root mean square error
SSGAN	Secure steganography based on generative adversarial network
StackGAN	Stacked generative adversarial network
SRGAN	Super resolution generative adversarial network
SSIM	Structural similarity index measure

Boriane Y. Tchaleu is with the School of Electrical and Electronic Engineering, Faculty of Engineering and Built Environment, University of Johannesburg, Corner Kingsway and University Road, Auckland Park, 2006, South Africa.

Alain R. Ndjiongue is with the School of Electrical and Information Engineering, Faculty of Engineering and the Built Environment, University of the Witwatersrand, 1 Jan Smuts Ave, Braamfontein, Johannesburg, 2017.

Collins A. Leke is with the School of Accounting at the University of Johannesburg, D-Ring 607 Auckland Park Kingsway Campus, Johannesburg, South Africa.

StyleGAN	Style generative adversarial network
VAE	Variational autoencoder
WGAN	Wasserstein generative adversarial network

I. Introduction

Over the recent years, a cutting-edge class of deep learning models have been developed, namely generative adversarial network (GAN). The initial GAN was proposed by Ian Goodfellow *et al.* as an innovative framework for unsupervised learning [1]. In this framework, two neural networks are perfectly trained to compete against each other. The popularity and recognition of GANs in real-world applications have lately increased due to their effectiveness and performance when trained with complex and high-dimensional data. These algorithms can be used in domains such as human image generations [2], [3], image-to-image translation [4]–[8], text-to-image generation [9]–[11], natural language processing [12]–[14], and computer vision [15]–[18], to mention only a few. GANs have exceptional advantages over other generative models, such as deep generative models (DGMs) and variational autoencoders (VAEs).

DGMs are algorithms that can generate new data from previously learned patterns [19]. These models include generative probability networks, deep belief networks, restricted Boltzmann machines, and deep Boltzmann machines. The models work well when managing large amounts of data. Traditional VAEs [20], [21] perform analysis, classification, or prediction based on inputs and rules. Furthermore, training them requires a significant amount of data and computational resources [22], [23]. In addition to being complex and opaque, they make predictions in a manner that is difficult to understand. As a result, it can be challenging to ensure that the model provides fair and unbiased decisions. Thoroughly, DGM and VAE samples are ambiguous, imprecise, and computationally demanding, especially when derived from high-dimensional Markov chain models [24]–[26]. To address the above challenges, Goodfellow *et al.* created GANs, which provide a different approach to dealing with Markov chain and Monte Carlo problems occurring during back-propagation, by including discriminative and generative models to the GAN training process [1]. This feature is advantageous when using GANs to generate realistic images because it avoids the complexity associated with maximum likelihood learning [27].

In GANs, the generator and discriminator models are derived from the two-player game theory [28]. While the generator algorithm generates data identical to real data, the discriminator algorithm distinguishes between created and actual data. The discriminator aims to reduce the generator score, while the generator seeks to maximize its score, making it a zero-sum or min-max game. The min-max game theory can be presented as a strategy to minimize the maximum possible loss that results from a player's choice. The purpose of training a GAN is to find a proper Nash equilibrium, where neither player can raise their score by changing their strategy [1]. The generator's main goal is to create as much fake data as possible by learning the characteristics of the probability distribution of real data. On another front, the discriminator determines the likelihood of the sample provided by the training set. An effective discriminator will produce samples assigned to the scalar value of 1, while the generator will produce samples assigned to 0. The generator attempts to extract the most important pattern from a random noise vector, creates a sample, and sends it to the discriminator. The goal is to fool the discriminator into thinking that the sample originates from the training set and produces a probability of 1. Without explicitly assuming a particular type of probability distribution, GANs can produce samples that correspond to the true distribution through this adversarial training [29], [30].

Several recent papers focused on architecture and methodology of generative models. GANs, being generative models, come with many variants and can be used in various domains. A limited number of GAN reviews are available in the literature. A list of examples is provided below. The authors of [31]–[33] describe the operation of GANs and the latest GANs, while a basic overview of certain GAN models can be found in [34]–[37], and [38] and [39] discuss GAN development patterns and their relationship with parallel intelligence. In [40], various GAN methods are examined from the point of view of algorithms and applications. The authors of [41]–[43] provide a comprehensive explanation of the variants of GANs with advantages and disadvantages, while in [44], the authors propose improving techniques that can improve semi-supervised learning performance and improve sample generation. Furthermore, [45] and [46] propose improved techniques for training single-image GANs using, for example, Wasserstein GAN (WGAN).

Given the brief introduction of GAN architecture and the literature overview above, it is believed that a comprehensive review is needed to touch on GANs and their variants to keep researchers, industry, and other readers informed of the evolution happening in the field. Consequently, the contributions of this paper can be summarized as follows:

- We discuss improvements in GAN designs and algorithms by exploring the fundamentals and history of GANs. We also review the applications of GANs in research to solve problems with unsupervised learning models.
- We explain and provide clarification on some GAN approaches with examples. This is to all the more likely comprehend the contrast between GAN models and variations.
- We offer a thorough review of the diverse world of GAN versions that have been suggested, each tailored for specific tasks and their shortcomings.
- We further explain the optimization strategies put forth

to address GAN problems. Optimization metrics have advantages such as improved modeling ability and extraction of deeper features for image tasks.

- We provide a meta-review of the history of GANs and the different applications using graphs.
- After reviewing application domains within most designs with examples from real-world situations, we offer interesting research prospects in this fast-expanding topic.

In the remainder of the paper, we discuss the GAN architectures, including diagrams in Section 2. In Section 3, we discuss and explain some GAN approaches, and provide a comprehensive explanation of various GAN versions in Section 4. In Section 5, we elaborate on the problems and discuss solutions. In Sections 6 and 7, we provide information on how the chosen model is applied in real-world scenarios with examples. We elaborate on some performance evaluation metrics across various sectors in Section 8. In Section 9, different graphs are provided to summarize the previous section. After that, we propose significant avenues for future research and provide closing thoughts.

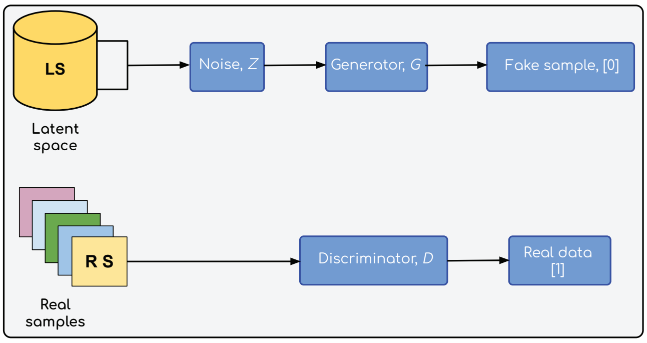


Fig. 1. Generator and discriminator GANs building blocks. The discriminator receives random input noise and generates fake samples (0), while the discriminator outputs real data from real sample data [47], [48].

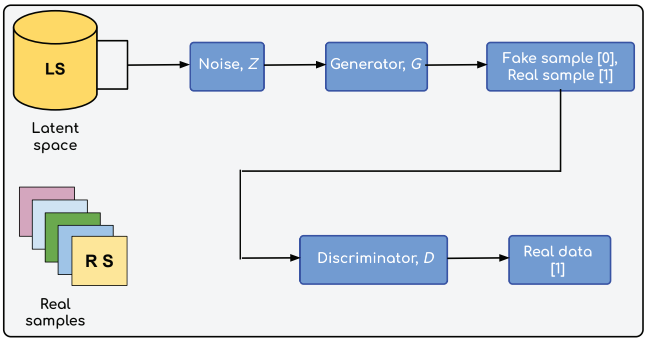


Fig. 2. Generator and discriminator GANs building blocks. The discriminator recognizes the data created by the generator as real data (1), whereas the generator generates data points that are recognized as fake data (0) [47], [48].

II. GAN Architecture

A generative model is designed primarily to learn an unknown probability distribution, which is used to sample the training observations. Sampling can be performed on newly generated data based on the training distribution after the model has been effectively trained. Two separate networks make up the GANs architecture: the generator and the discriminator. While the generator creates synthetic samples from a given noise z, the discriminator takes samples and actual data as input. To determine whether a sample originates from the generator or from actual data, the discriminator computes the probability that each sample originates from the real dataset. The generator exploits the information from the outcome of the discriminator to determine how to best optimize its parameters in the subsequent iteration. To achieve their respective objectives, G and D compete against each other. The samples generated by the generator are termed fake samples, as demonstrated in Figs. 1 and 2. The discriminator recognizes a data point given as input from the training dataset as a real sample (output a scalar value of 1), while the data point generated by the generator is recognized as a fake (output a scalar value of 0). The adversarial relationship between the generator and the discriminator lends a distinctive elegance to this approach. The generator generates samples such that the discriminator misidentifies one as real, hoping to trick it into thinking that it is a real sample when it receives a fake sample [47]. In summary, the generator tries to fool the discriminator.

Taking into account both the optimization process and the objective function, in the Min-Max optimization formulation, the generator aims at minimizing the objective function, while the discriminator aims at maximizing it, as illustrated in Figs. 3 and 4. These can be mathematically expressed as

$$\begin{aligned} &min_{G}max_{D}V(D,G) = [E_x \sim P_d(x)[\log(D(x))] \\ &+ E_z \sim P_z(z)[\log(1 - D(G(z)))]], \end{aligned} \tag{1}$$

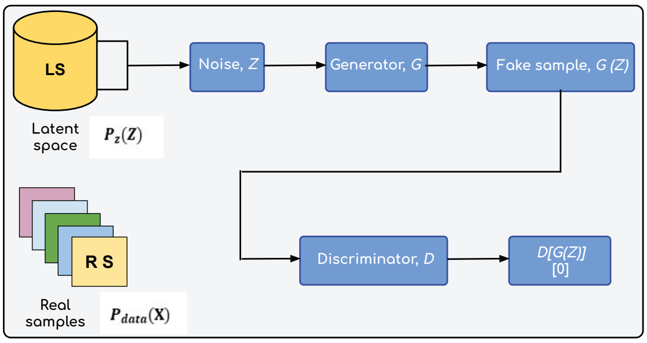


Fig. 3. GANs objective function. The output sample from the generator is optimized and the discriminator generates $D(G(z)[0])$.

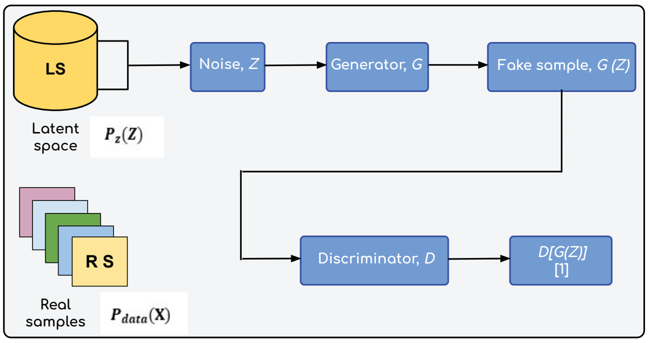


Fig. 4. GANs objective function. The objective function is optimized to $D(G(z)[1])$ by the discriminator.

where the probability distribution of the latent space, P_z, is often a random Gaussian distribution [49]. The probability distribution of the training dataset is denoted by P_d. The discriminator seeks to classify the sample taken from $P_d(x)$ as a real sample, while the generated sample is $G(z)$. When the discriminator receives $G(z)$ as input, it attempts to identify it as a fake sample. It aspires to reduce $D(G(z))$ to a value of 0. It seeks to maximize $[1 - D(G(z))]$, while the generator looks ahead to make the discriminator mistakenly identify a created sample as real by forcing the likelihood of $D(G(z))$ to 1 as illustrated in Fig. 4. Hence, the goal of the generator is to minimize $[1 - D(G(z)]$. Thanks to the development and evolution of GANs, generative models are now capable of producing realistic images with encouraging outcomes.

III. GAN Approaches

This section provides a detailed explanation of GAN-based approaches with some examples. This is to better understand the difference between GANs and traditional generative models.

A. *Text-to-text GAN (TextGAN)*

An objective of text generation is to create text that appears to be indistinguishable from text written by a human. It is also called natural language generation. Although large language models are successful in creating text token-by-token by trying to make each next word meaningful in the given context, GANs overcome large language models because of their ability to generate a whole sample of text as one output. Before the invention of GANs in 2014, there have been numerous attempts to solve text generation problems. GAN algorithm, naturally suited to the continuous image type of data, had a lot to struggle with the discrete type as text. The development of GAN-based approaches has made it possible to produce high quality text today [50].

B. *Leak GAN (LeakGAN)*

LeakGANs were created in 2018 by Guo *et al.* LeakGANs can generate long, coherent, semantically relevant text by leaking information from a FeUdal network module-based generator and a feature extraction discriminator. The LeakGAN training system is based on a neural network architecture that utilizes the generator-discriminator model and includes a generator module composed of networks such as (*i*) a generator-worker module with short- and long-term memory for generating action embedding vectors and (*ii*) a long short-term memory generator-worker module, which generates a feature subgoal vector and is then linearly projected into an action subgoal. Furthermore, it is constructed using a discriminator network, which consists of a convolutional neural network that uses a feature extractor and softmax classification to extract information that has been leaked [51]–[54].

C. *Sequence Generation GAN (SeqGAN)*

A SeqGAN is a GAN dedicated to sequence generation. For example, the algorithm can be used for a variety of real-world tasks, including the composition of poems, the generation of speech, and the creation of music. It is a GAN made up of discriminators and generators. The gradient policy update is performed directly by SeqGAN. The generator differentiation problem is circumvented in this method by modeling the data generator as a stochastic policy in reinforcement learning. After being evaluated on an entire sequence by the GAN discriminator, the reinforcement learning reward signal is returned via a Monte Carlo search to intermediate state action phases [55].

D. *RankGAN*

RankGAN was introduced as a new stage-wise learning paradigm for training GANs with the idea of strengthening the discriminator and thus the generator in each stage [56]. The novelty of the work lies in the RankGAN architecture, which consists of a discriminator that rates the quality of images produced by several generating stages. The ranker helps the generators to pick up the finer points in the training set so that they can get better at each level. In the literature, RankGANs is viewed as a maximum margin GAN ranking for face generation compared to WGAN and least squares GAN (LSGAN) models [57].

E. *MaskGAN*

GANs that use context-based masking to fill in missing text are known as MaskGANs. Their foundation is a sequence-to-sequence neural network architecture, assessed through the MaskGAN benchmark task, and trained using the MaskGAN training system. Furthermore, it is composed of the following: (*i*) The generator neural network. Using the encoder's hidden states, the generator-decoder fills in the missing tokens and decodes the masked sequences; (*ii*) The discriminator neural network. The discriminator neural architecture is the same as that of the generator network. It takes as inputs the original context and the generator

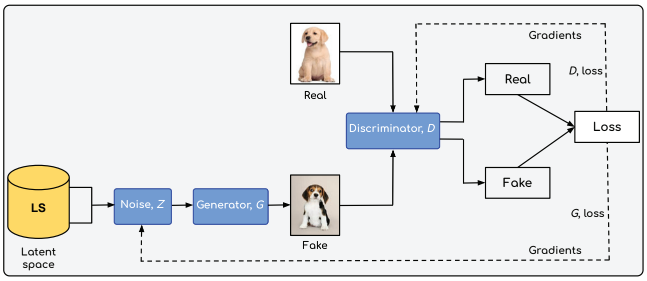


Fig. 5. Vanilla GAN: The generator generates outputs and the discriminator. Identifies the input as real or fake and tries to minimize the loss (adapted from [59, Fig. 10]).

filled-in sequence. Given the actual context of the masked sequence, it calculates the probability that each token is real; (*iii*) A critic neural network. It calculates the discounted total return of the filled-in sequence and is applied as an extra head-off to the discriminator. Both generator and discriminator networks consist of an encoding-decoding module [58].

IV. GAN Variants

This section elaborates on the most popular GAN variants available in the literature.

A. Vanilla GAN

One of the most basic GAN forms is Vanilla GAN. As depicted in Fig. 5, Vanilla GAN is made up of the generator and the discriminator. Multilayer perception is used internally by the generator and discriminator to create and categorize images. While the generator captures data distribution, the discriminator attempts to determine whether the input belongs to a particular category. The process of minimizing the loss occurs when the feedback is provided to the generator and discriminator, after the computation of the loss function [3], [59]–[61].

B. Conditional GAN (cGAN)

cGAN is a revolutionary approach to text-to-image synthesis and image generation. As shown in Fig. 6, cGAN is composed of a generator and a discriminator. They are given additional information, which can be labeled or unlabeled data, and their role is to assist in the accuracy of prediction. Moreover, unlike the joint probability, in cGAN, the discriminator determines the conditional probability. During the training process, the images are passed to the network with actual labels to distinguish them. However, less labeled data is required when using cGANs. cGANs can handle complex data and ask the model which image to generate [62]–[64].

For the generator to produce a sample, it needs a noise vector. Nevertheless, a conditional generation also requires auxiliary data that instruct the generator which class of sample to generate. Given the conditioning label y, as an example, the generator creates fake information by combining y with the noise vector z as

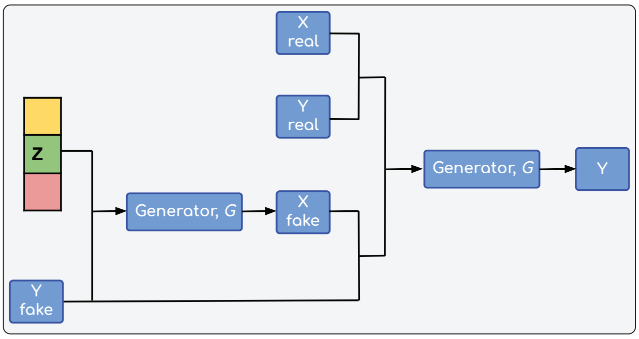


Fig. 6. Conditional generative adversarial network. The generator receives labeled or unlabeled data, which aids in the discriminator's ability to make accurate predictions (adapted from [65, Fig. 3(b)]).

$$G(z, y) = (x|y), \tag{2}$$

where x is the fake information generated. From (2), it can be seen that x is conditional on y. In order to fool the discriminator, the fake sample attempts to imitate the real sample for the provided label as closely as possible. It is therefore essential that the generated instances match the generator's label to ensure that realistic data is produced.

C. Auxiliary Classifier GAN (ACGAN)

Depending on the application, several cGAN extensions can be used. An illustration is ACGAN, which uses a discriminator to predict the class label of an image rather than just using it as input. When learning a representation in the latent space independent of the class labeled, it stabilizes the training process and enables the creation of huge and high-quality images [66]–[68]. ACGAN architecture consists of a discriminator to categorize generated images as real or fake and a generator model to generate images using random points from a latent space as input. Then, in a zero-sum game, both models are taught simultaneously, as illustrated in 7.

D. Deep Convolutional GAN (DCGAN)

DCGAN is one of the most popular and effective GAN variants. This is because it is the first variant of GANs that utilizes a deep convolutional network as a generator, producing high-resolution and high-quality images for differentiation and eliminating connected layers. DCGAN uses convolutional and convolutional-transposed layers. DCGAN is illustrated in Fig. 8. The generator employs *Leaky-ReLu* activation for all layers and *Tanh* activation for the final layer. In addition, both the generator and the discriminator use batch normalization for the output and input layers, respectively. The discriminator uses all layers employing the

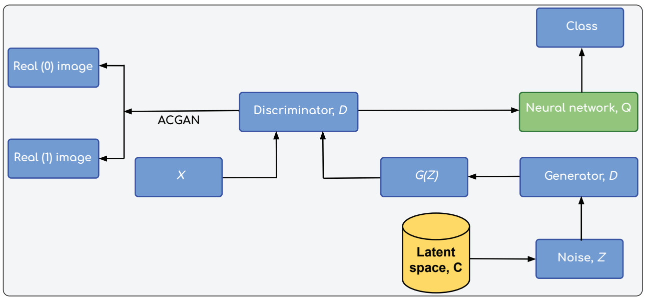


Fig. 7. Auxiliary classier generative adversarial network. Instead of accepting an image as input, the discriminator is changed to use a neural network to predict the class label of a given image (adapted from [65, Fig. 3(f)]).

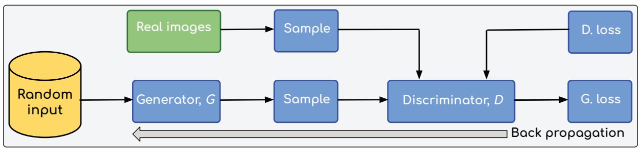


Fig. 8. Deep convolutional GAN. back-propagation is utilized by the discriminator and generator in conjunction with convolutional and convolutional-transpose layers (adapted from [69, Fig. 5]).

Leaky-ReLu activation function. Optimization is done with the *Adam optimizer* at a specific learning rate. Furthermore, DCGAN utilizes transposed convolution for up-sampling [4].

E. Stacked GAN (StackGAN)

StackGANs have two stages, namely Stage I GAN and Stage II GAN, and use two generators and two discriminators at each stage [5]. The Stack-I GAN utilizes encoders for text, conditioning augmentation networks, generators and discriminators, and embedded compressor networks. However, Stack-II GAN exploits a text encoder, a conditioning enhancement network, both generator and discriminator networks, and a built-in compressor network. Overall, stacking multiple generators pushes the limits of the StackGANs' capabilities and creates more realistic and high-resolution images, as shown in Fig. 9. Figure 10 illustrates Stages I and II used to generate high-resolution images of 256 × 256 pixels and low-resolution images of 64 × 64 pixels. The embedding compressor network, which uses pre-trained character level embedding, transforms the variable-length text input into a vector with a fixed length and then applies the condition augmentation network and residual blocks.

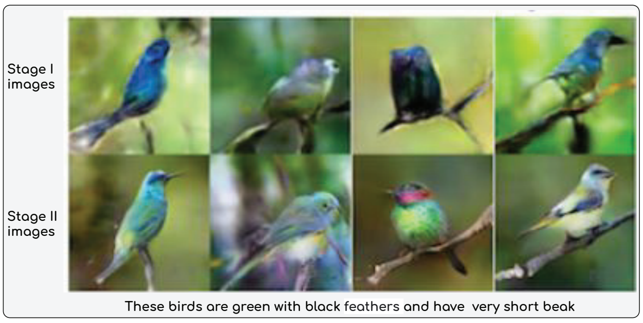

Stage I images
Stage II images
These birds are green with black feathers and have very short beak

Fig. 9. StackGAN stages examples. Examples of a GAN model using two generators and two discriminators (adapted from [70, Fig. 10]).

In 2017, H. Zhang *et al.* introduced an improved version of StackGAN called StackGAN++, also known as StackGAN-V2. The design of StackGAN++ differs from StackGAN in 2 major ways, namely: (*i*) Multiscale image synthesis: Each branch of the architecture, which resembles a tree, represents a different generator network. The higher the branch size, the larger the resulting image gets; (*ii*) Unconditional loss: By jointly approximating multiple distributions, StackGAN++ has a more stable training behavior than simple StackGAN [70]–[72].

F. Wasserstein GAN (WGAN)

WGAN was developed in 2017 with the stated goal of improving GANs learning stability [73]. Figure 11 depicts the WGAN architecture. As in simple GANs and unlike Jensen-Shannon divergence, WGAN minimizes the approximation of the earth mover's distance [8]. Training is more stable and shows less evidence of mode collapse than with the original GANs. Additionally, helpful curves for troubleshooting and finding hyperparameters are produced. WGAN modifies or replaces the discriminator network with a critic that rates a given image's truthfulness or falsity. In contrast, this approach does not use a discriminator to forecast the probability of the generated images being real or fake [74]. Furthermore, gradient descent is used in training the generator, which differentiates it from simple GANs as the discriminator is now restricted to having a bound Lipschitz norm. In WGAN training, the upper bounds of Lipschitz norms are applied in conjunction with weight clipping, whereby the discriminator is implemented through a multi-layer perceptron [73]. Moreover, adding spectral normalization to WGAN's training allows rapid convergence [75]. An improvement in WGANs can be obtained by avoiding strictly binding the discriminator and adding a penalty gradient, which would speed up the training process [46].

G. Information Maximizing GAN (InfoGAN)

An InfoGAN is an illustration of a GAN that modifies their purpose to promote the development of meaningful and understandable representations. In InfoGAN, optimization is maximized by mutual pieces of information between the observations and a fixed small subset of the noise variables [76], [77]. The goal function of the original GAN is simply modified to include a regularization term to implement InfoGAN, as shown in (3) and Fig. 12.

$$min_G max_D V_I(D|G) = V(D|G) - [\lambda I(c : G(z|c))]. \tag{3}$$

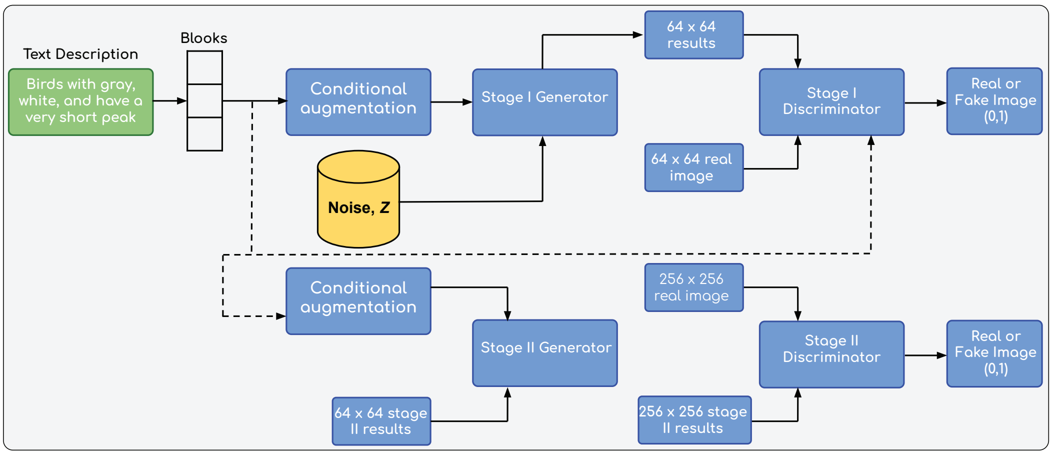


Fig. 10. StackGAN model architecture with stage I and stage II. High-resolution and low-resolution images are produced by the generator and discriminator (adapted from [70, Fig. 1]).

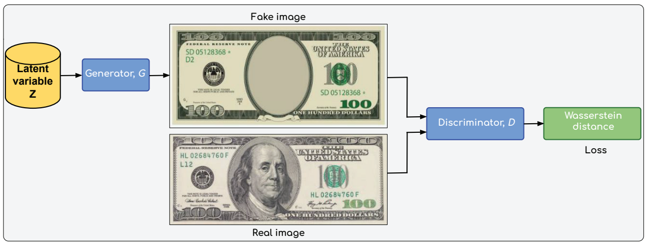


Fig. 11. Wasserstein generative adversarial network architecture. The generator generates real and fake money images (adapted from [46, Fig. 1]).

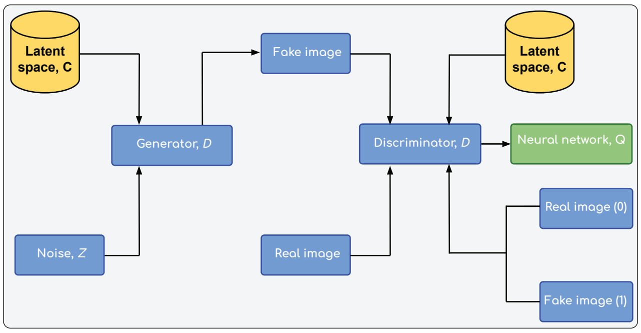


Fig. 12. Information maximizing GAN. A neural network is used to modify the standard GAN model to promote the development of meaningful and comprehensible representations (adapted from [77, Fig. 1]).

In this equation, the regularization constant, λ, is commonly set to unity and $[I(c : G(z|c))]$ refers to the mutual information between the latent code c and the generator output $G(z|c)$.

H. Least Square GAN (LSGAN)

In LSGANs, the discriminator utilizes the least square loss function. The Pearson divergence is reduced by minimizing the objective function, which is defined as [78]

$$\begin{aligned} min_D V_L(D) = \frac{1}{2}\exp_x \sim p_d(x)[(D(x)-b)^2] \\ +\frac{1}{2}\exp_z \sim p_z(z)[(D(G_{(z)})-a)^2], \end{aligned} \tag{4}$$

which can be reduced to

$$min_G V_L(G) = \frac{1}{2}\exp_z \sim p_z(z)[(D(G(z))-c)^2], \tag{5}$$

Equations (4) and (5) explain the objective function. The labels for fake and actual data are named a and b respectively. This indicates the value that the generator wants the discriminator to attribute to the fake data. In LSGANs, the least squares objective function is exploited in training both the generator and discriminator. The least-square loss penalizes samples about their position concerning the decision boundary more severely than the sigmoid cross entropy.

I. Dual Video Discriminator GAN (DVD-GAN)

DVD-GAN is a GAN variant for video generation built upon a big generative network (BigGAN). BigGAN uses deep learning techniques to create high-quality and realistic images. This variant of GANs is made up of two discriminators, namely a spatial and a temporal discriminator. By selecting k full-resolution frames randomly and evaluating each separately, the temporal discriminator evaluates the content and structure of a single frame. The learning signal creates movement and provides Gaussian latent noise, z, via the temporal discriminator [79].

J. Progressive GANs (PGAN)

Figure 13 illustrates PGAN, another variant of GAN. In this case, the earliest layers of the generator create images with very low resolution, and later layers add details. In PGAN, the training process is faster than equivalent non-progressive GANs and generates images with higher resolution [80].

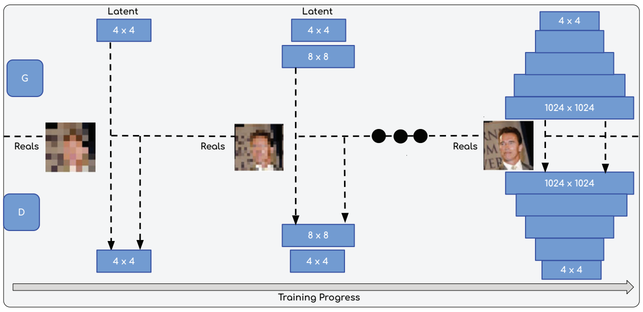


Fig. 13. Progressive GAN: The generator's initial layers generate very low-resolution images, while its later layers generate high-resolution images (adapted from [65, Fig. 5]).

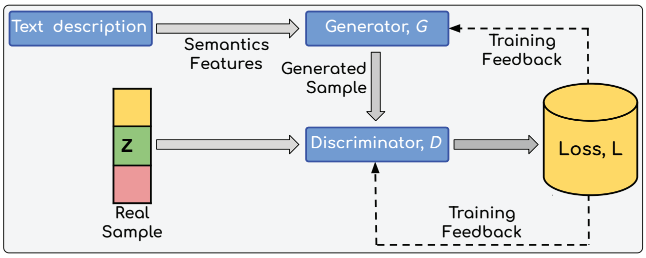


Fig. 14. Text-to-image synthesis generative adversarial network. *D* differentiate inputs based on the features specified in the text description, while *G* attempts to generate based on the description (adapted from [69, Fig. 1]).

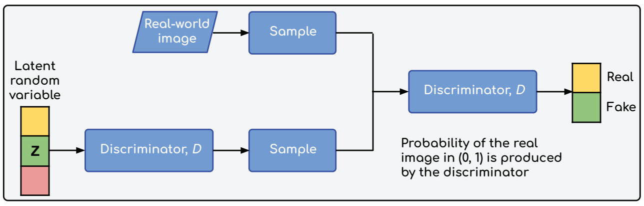


Fig. 15. StyleGAN train jointly in a mini-max game. A reference photo is used by GAN to provide efficient results and improve the generator (adapted from [81, Fig. 1]).

K. Generative Adversarial Text-to-Image Synthesis

Generative adversarial text-to-image synthesis can identify an image from the dataset that most closely resembles the text description and produce comparable images. According to the description, the generator generates data, and the discriminator makes distinctions based on the attributes listed in the text description. A 100-dimensional noise vector sampled from a latent space is obtained by combining a random normal distribution with the encoding of the textual description in 256 dimensions, as shown inFig. 14. This approach helps the generator create images that match the specified description, instead of random images. Thus, a pair of images and text embeddings are sent to the discriminator as inputs rather than a single image. The output signals are 0 or 1. The discriminator's duty in this model is to determine if an image is authentic or fake. As a result, the generator is forced to produce images that correspond to the description provided in the text, as well as ones that seem realistic. A variety of different pairs (pictures or texts) are fed into the model during the training process to satisfy the dual responsibility of the discriminator. For the model to determine whether a given image or text pair is in alignment with one another, a pair of real and wrong image captions is provided. The incorrect image implies that there is a mismatch between the image and the caption. Thus the output variable is set to 0, so that the model can understand that the provided image and description are not aligned. In the case of real images with real captions are used as input, the target variable would be set to 1.

L. Style GAN (StyleGAN)

In contrast to other GANs which aim at enhancing the discriminator, StyleGANs enhance the generator. By using a reference photo, this GAN produces effective results. Figure 15 depicts the StyleGAN architecture, which includes a mapping network, whose role is to represent the input in an intermediate latent space followed by adaptive instance normalization (AdaIN) after each layer [81]. The evolution of StyleGAN led to the creation of different types of StyleGANs such as StyleGAN2, StyleGAN2-ADA, and StyleGAN3. All these upgraded versions remove some of the characteristic artifacts of StyleGAN and improve image quality.

$$\begin{aligned} min_{\theta g} max_{\theta d}[E_z \sim p_d \log D_{\theta_d(x)} \\ +E(_z\sim p(z)) \log(1 - D_{\theta_d}(G_{\theta g}(z)))]. \end{aligned} \tag{6}$$

Equation (6) shows that AdaIN works by receiving inputs for both content and style, namely x and z, and simply aligns the channel-wise mean and variance of x and z.

M. Cycle GAN (CycleGAN)

Figure 16 shows an illustration of CycleGAN, which maps one image to another, called image-to-image translations. Climate change is one of its applications. When the summer and winter seasons are translated from one image to another, a mapping function in CycleGAN can be used to convert pictures from summer to winter and vice versa. Features can then be added or removed following the mapping function, resulting in the least amount of loss between the predicted and actual outputs [82]–[84].

N. DiscoGAN

DiscoGAN is a GAN that generates images of products in a domain given an image in a different domain. It has gained popularity recently due to its capacity to identify cross-domain relationships from unsupervised data. Cross-domain relationships are extremely natural for humans. Humans can determine the relationship between two separate domains from their visual representations. As illustrated in Fig.

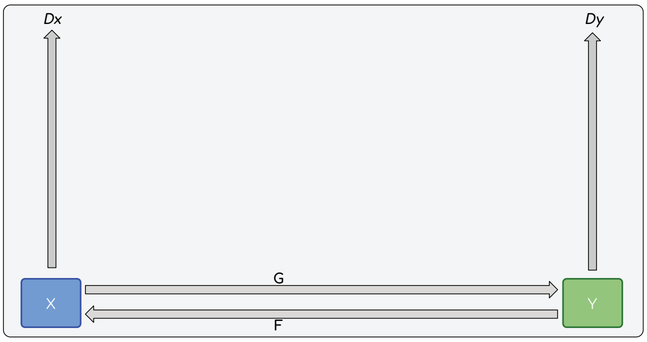


Fig. 16. CycleGAN using GANs to map one image to another (adapted from [85, Fig. 1]).

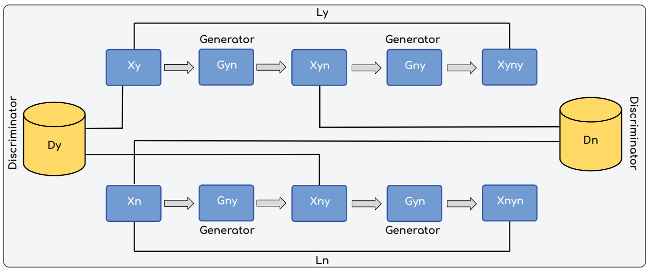


Fig. 17. Disco generative adversarial network. Using loss functions, an image is converted from one domain to a different one and back again to create images that resemble the original (adapted from [87, Fig. 4.3]).

17, it contains photos from two separate domains. Upon first inspection, it is clear that they are associated due to their exterior color. It is incredibly challenging to create a machine learning (ML) model that identifies such a linkage given unpaired photos from two separate domains. DiscoGAN recently demonstrated encouraging results in learning such a connection across two distinct domains [86]. Like CycleGAN, DiscoGAN is also built on the fundamental of reconstructing loss functions as illustrated in Fig. 17. The concept is that the created image should resemble the original as much as possible when translated from one domain to another. It should also return to the original one. In this instance, the model attempts to minimize reconstruction loss during training, defined as the quantitative difference.

O. Pixel-to-Pixel GAN (Pix2Pix GAN)

A GAN model named Pix2Pix was created specifically to transform one image into another. For example, the conversion of a daytime landscape photo to a nighttime one. Its foundation is cGAN. It generates a target picture dependent on an input image. As a result of its modification from cGAN, the Pix2Pix GAN generates an image that is plausible in both domains. This is a translation of the input image and a suitable representation of the target domain content. The U-Net generator model, patchGAN discriminator model, and model optimization process are meticulously specified in the Pix2Pix GAN architecture. Convolution-batch *normalization-ReLu* blocks of layers are commonly used in deep convolutional neural networks and are the case for both the generator and discriminator models. Notably, image-to-image translation is a challenging operation that often necessitates specific models and loss functions for each translation dataset [88], [89].

P. Big GAN (BigGAN)

BigGAN is another GAN variant that is among the top models on the market because of its large-scale and effective imaging capabilities [74]. To train big networks, this model uses additional parameters, which boost model performance greatly and yield extremely detailed results. Controllability and the initiation score applied to the model output are two significant characteristics of BigGAN that demonstrate the superior performance of GAN versions. Increasing the unconditional image processing power of BigGAN helps to increase performance. Furthermore, the Frecher inception distance (FID) and inception score (IS) are also used in the evaluation to compare the model to state-of-the-art models.

Q. Super Resolution GAN (SRGAN)

SRGAN is a modified GAN model that has been developed to synthesize high-resolution images from GAN [18]. However, SRGAN has two limitations: *(i)* Since the model accepts an input image with a lower resolution, SRGAN cannot be applied to the sample generation task. It is therefore not a generative model, unlike the original GAN, which uses noise variables as its inputs. *(ii)* Higher-resolution samples are still needed for SRGAN training. For example, to synthesize 128 × 128 resolution images from 64 × 64 resolution input images, SRGAN needs to be trained on a training set of 128 × 128 resolution images. Consequently, when a lower resolution ground-truth training set is provided, like the 32 × 32 resolution images in the CIFAR-10 dataset, SRGAN cannot synthesize 128 × 128 quality images. Both the generator and discriminator use convolutional layers, as seen in Fig. 18, the discriminator employs the *Leaky-ReLu* activation function, while the generator uses the *Parametric ReLu*. Using the input data, the generator calculates the average of all formulated potential solutions, down to the level of pixels.

R. Laplacian Pyramid GAN (LAPGAN)

LAPGAN is a linear invertible image representation that consists of a set of band-pass images spaced an octave apart with a low frequency. It generates very high-quality images by utilizing many tiers of the *Laplacian pyramid* and numerous generator and discriminator networks. The image is first down-sampled at each tier of the pyramid, as seen in Fig. 19. After that, it undergoes a backward pass at these layers, where it is upscaled and gains noise from the conditional GAN layers, until it returns to its initial size [90], [91].

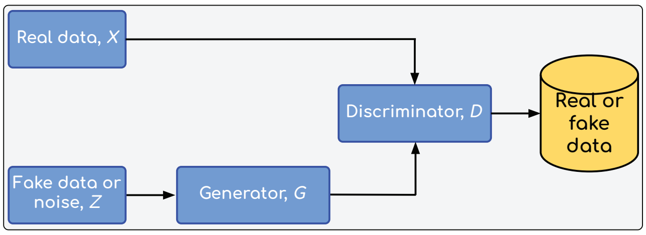


Fig. 18. Single Image SRGAN. To convert low-quality images into higher-resolution images, the discriminator and generator networks both use convolutional layers (adapted from [65, Fig. 3(a)]).

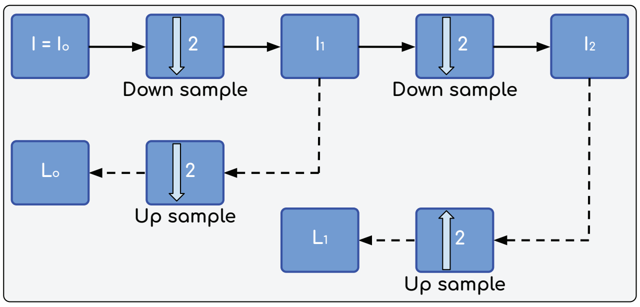


Fig. 19. Laplacian pyramid generative adversarial network. Numerous generator and discriminator networks, as well as different laplacian pyramid levels, are used to produce high-quality images (adapted from [65, Fig. 4]).

S. Energy-based GAN (EBGAN)

EBGAN model sees the discriminator as an energy function that assigns higher energies to some regions and lower energies to those close to the data manifold. A generator is trained to generate contrasting samples with low energies, much like the probabilistic GANs. A discriminator is trained to assign high energies to these generated samples. When the discriminator is viewed as an energy function, a variety of architectures and loss functions may be used in addition to the traditional binary classifier with logistic output [92].

T. Mode Regularized GAN (MRGAN)

MRGAN is a different GAN version that encourages the generator to generate images from every mode of data distribution. A penalty function named mode regularizer encourages the generator to generate images near the data distribution modes. [65].

V. GAN Challenges and Solutions

More often, the performance of GANs is prone to random inconsistencies. The majority of these issues are related to their training and are an active area of research. The following paragraphs describe these challenges and suggest solutions to mitigate them.

A. Mode Collapse

Mode collapse is one of the GAN models' main challenges. This issue typically arises when the generator learns to create only one type of output, a small set of outputs, or identical results, disregarding the input. This occurs during the training process when the generator identifies a specific type of data that is easily deceived by the discriminator and keeps on generating that specific type. As the generator runs longer, the output quality declines and the process becomes unstable and ineffective. Then, the entire system over-optimizes on that one output as the generator has little motivation to change things. Multi-modal data distributions indicate that each sample is often assigned to a single label. Ten types of digits or modes designated from 0 to 9, for instance, are present in the modified National Institute of Standards and Technology Database (MNIST). Figure 20 demonstrates MNIST images produced by two distinct GANs. A ”good” GAN that has not experienced mode collapse is trained in the top row. Throughout the training, it generates all possible modes. Only the number ”6” is produced by the GAN training with mode collapse in the bottom row [93], [94].

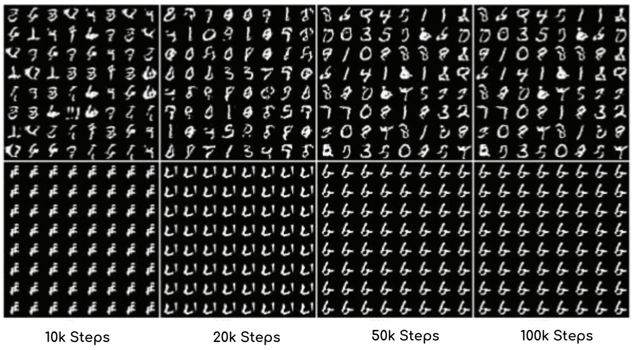


Fig. 20. Mode collapse generating digits of 8s repeatedly (adapted from [95, Fig. 3]).

A mode collapse is easily identified when the loss for the generator and the discriminator keep oscillating over time and also when the generator model is successful in producing similar output images from various points in the latent space. This is shown in the literature when using 100 samples of the MNIST dataset [95]. Even though the loss for the generator jumps up and down, the loss for the discriminator remains within a reasonable range. The discriminator accuracy also shows a higher number frequently around 100%, indicating that it has excellent ability to distinguish between genuine and fraudulent (real and fake) instances over multiple batches, which is not good for image quality or variety. In all, a mode collapse can be easily identified when the loss for the generator and the discriminator keep oscillating over time. This also happens when the generator model is successful in producing similar output images from various points in the latent space. Several techniques with enhanced network architecture and new objective functions or alternative optimization techniques have been presented in recent works to address this problem [95]–[97]. It should be noted that there is no right way of preventing stable GANs from suffering from mode collapse. But, some examples have been demonstrated. In the following paragraphs, we discuss a few techniques to prevent mode collapse.

Four main techniques are available to prevent mode collapse in GANs: (*i*) The limit latent space size. The most reliable approach might be to constrain the latent dimension size. By doing so, it forces the model to only produce a restricted or small subset of plausible output; (*ii*) Qualitative measures. Manually reviewing images works better when the problem is obvious. In more complicated cases or with significant amounts of data, it might not work as well; (*iii*) Quantitative measures. Quantitative measures using specific sets of object classes like FID or IS rely on pre-trained models. They do not have an upper bound because the maximum score is theoretically infinite [96]; (*iv*) General measures. GAN variants that encourage the generator to create a variety of realistic outputs, such as mini-batch discrimination, unrolled GANs, or WGANs, help to prevent mode collapse [95]. Furthermore, mode collapse is prevented during training by introducing noise into the discriminator and increases efficiency.

B. Memory and Computational Constraints

Another GAN challenge is memory and computational constraints that limit the size and complexity of the models. Domains like image translation in medicine, photo translations, and 3D models to name only three, use ML techniques such as large neural networks and deep learning. These techniques require lots of memory and computational power to be processed. Also, during the training process, the model takes a considerable amount of time and is costly.

The applications of techniques such as progressive growing and feature matching help reduce memory usage. Furthermore, the models' performance and efficiency can be increased by utilizing other GAN types, such as cGANs.

C. Evaluation and Optimization

An important question that arises while using GANs is how to evaluate and optimize generative models. A generative model differs from a supervised learning model in that in a generative model, a predefined metric or loss function is used to estimate the model's accuracy and quality. Because GAN models are unsupervised learning, there is no clear objective or ground truth to compare the output with. The output data has not been compared, and it is challenging to evaluate and optimize the process during the training phase.

To optimize and evaluate the models, techniques such as FID and IS [96], or perceptual similarity techniques can be used. As a result of these techniques, the output can be well examined, and qualitative and quantitative measures of output can be provided.

D. Data Availability and Quality

One issue with GAN models, which is not limited to generative models, is the lack of readily available and high-quality data. Generating complicated and high-resolution outputs from medical photos or 3D models, for instance, can be challenging because a sizable dataset of real data is required. This dataset can be costly and difficult to obtain. Additionally, the data must be clean, consistent, and relevant, which can be challenging in some domains, such as medical imaging, where the data provided is often noisy, incomplete, or sensitive.

Techniques like data augmentation, semi-supervised learning, and domain adaptation that help increase or improve the quality of the data and make it constantly available should be utilized to reduce inequality and data unavailability [97].

E. Ethical and Social Implications

The moral and societal ramifications of the outputs produced by GAN models present another challenge. While different versions of GANs can be used to create stunning output images in a variety of fields, including 3D models and the medical industry, they can also produce destructive and deceptive outputs, including false photos, fake news, and deepfake or synthetic identities. These results pose a threat to democracy, trust, privacy, and security. These issues raise a number of social and ethical concerns regarding the accountability and responsibility of GAN model developers and users. When using GANs, methods such as attribution or watermarking verification can be used to prevent and steer clear of social and ethical problems. The techniques are great for locating and safeguarding outputs [98].

F. Batch Normalization

The stability of convergence in GAN models can be prevented using batch normalization. The use of batch normalization techniques such as deep convolutional neural networks on GANs, such as deep convolutional neural networks, helps to stabilize the training process and avoid model instability.

VI. GANs Applications

This section explains how GANs can be used in a broad range of applications.

A. Generating Fake Human Faces using DCGANs

In image generation, the training process becomes unstable and ineffective when there are not enough images or data in the datasets. Data augmentation is a method that creates various versions of an image that already exists, resolving these lack of data issues in many domains. Traditional techniques and rotating, to mention only two, were employed before GAN models. But owing to

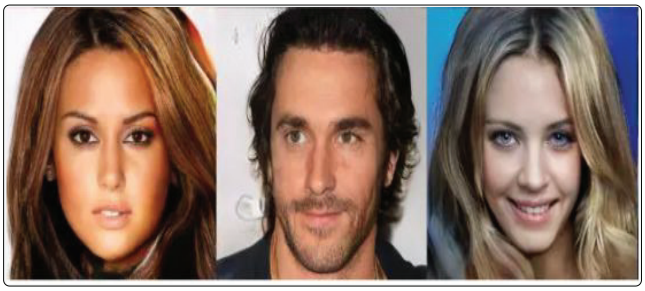


Fig. 21. Generation of human faces (adapted from [102, Fig. 1]).

GANs, new images can now be produced from pre-existing ones. Numerous tasks, such as high-fidelity natural picture production, data augmentation, and image reduction, have seen successful application of the technology [99]–[101]. DCGAN is the most often utilized GAN architecture in this application, where the model can be trained with the *CelebFaces Attributes* dataset.

B. Generating Images of Human Faces

In their 2017 work, Tero Karras *et al.* shows how to create realistic and convincing images of human faces [16]. The results are astounding and appear to be real. Furthermore, these models are successful in showcasing the ability to produce images of realistic-looking faces of people who do not exist (an example is the reviving Monalisa [102]). In Moscow, researchers from Samsung artificial intelligence (AI) used ML systems to validate this research [103]–[105], which represents an interesting application of GANs. None of the pictures in Fig. 21 are real. Some of the faces are pretty recognizable because the GAN models were trained using some celebrity faces as examples. GANs are appropriate for creating not only images but they can also be utilized to create scenarios and objects.

C. Generating Realistic Images

In [74] and [106], reports published in 2018, the authors describe how BigGAN techniques can create artificial images that are nearly identical to real images. In Fig. 22, we illustrate output images generated using such a variant of GANs.

D. Generating Cartoon Characters

In [107], published in 2017, Y. Jin *et al.* demonstrated how to train and apply a GAN to create animated character faces illustrated by Fig. 23. Many people have attempted to create Pokemon characters after being inspired by these examples, with varying degrees of success. Two examples of projects with limited success are pokeGAN and Pokemon generation using DCGAN [108]. With some complex designs, GANs have been claimed to produce extremely realistic anime characters. Examples of GAN designs applied in this case are DCGAN and StyleGAN.

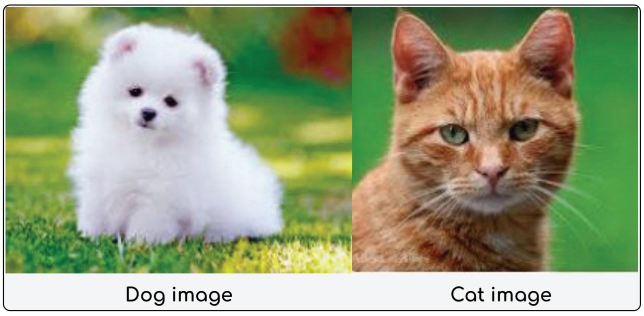


Fig. 22. Real image example extracted from large scale GAN training for high fidelity natural images and generated using BigGAN (adapted from [74, Fig. 2]).

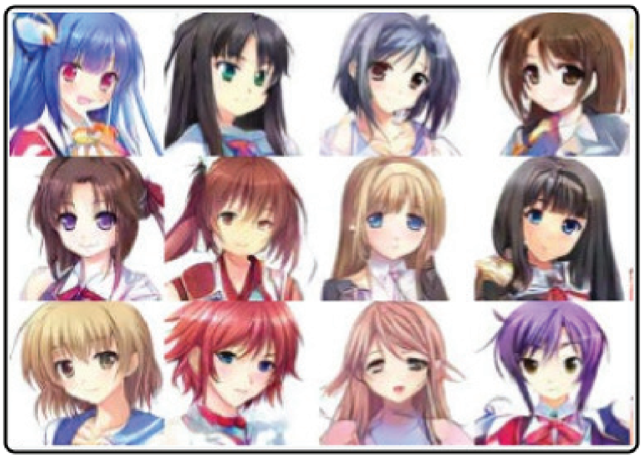


Fig. 23. Cartoon images implemented using conditional GANs (adapted from [107, Fig. 5]).

E. Translating from One Image to Another

Another interesting application for GANs is the image-to-image translation technique that exploits StyleGANs (the NVIDIA StyleGAN) and the Pix2Pix methodology. The conversion of semantic images into actual photograph structures, like buildings and skyscrapers, serves as a significant illustration of image-to-image translation. Additionally, it can translate satellite photographs to Google Maps. Apart from all these, old sketches and monochrome images can be converted to color ones. In addition to many other aspects, images can be translated from day to night and from semantic imagery to pictures of cityscapes [109]. A typical example is given by the authors of [82], where they explained how to translate horse into zebra and summer-to-winter as illustrated in Fig. 24. Another example is given by J. Zhu *et al.* who presented their well-known CycleGAN and several effective image-to-image translation examples [82].

F. Translating from Text-to-Image

Generating life-like visuals from written descriptions is a fascinating and demanding task. Generative adversarial models excel in this application by allowing the translation

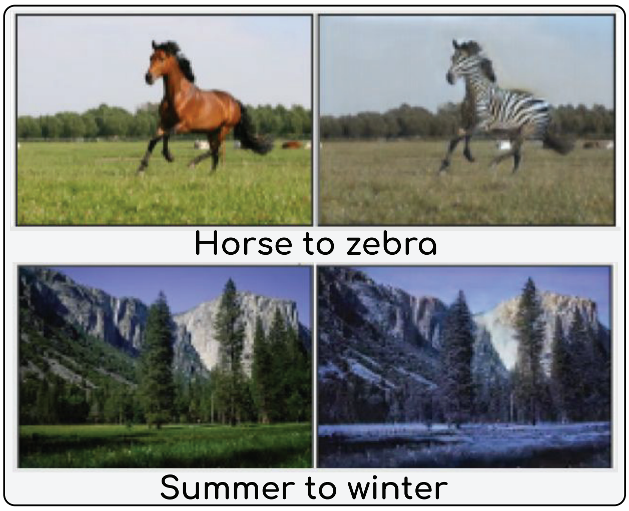


Fig. 24. Illustration of image-to-image conversion using cycle GANs (adapted from [1, Fig. 8]).

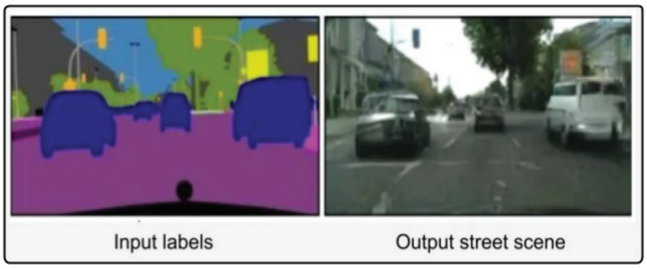


Fig. 25. An example of a semantic image to photo translation. (adapted from [111, Fig. 1]).

of text to an image through the use of StackGANs. As illustrated by Fig. 9, Han Z. *et al.* demonstrated how to use StackGAN to produce photographs that look realistic from textual descriptions of common objects like birds [110]. The only input provided for this task is a written description of the image, and the model is supposed to infer its assumptions based on several elements that may be unclear on the expectation. However, when it is necessary to create high-resolution photographs, details and elegance become an issue. The potential usage of this program in the future includes creating and/or developing products as well as describing various object types.

G. Translating from Semantic Images to Photos

According to [111], cGANs can be used to generate photo-realistic images based on an input semantic image or sketch. As illustrated in Fig. 25, images of a cityscape, a bedroom, or a person all conjure up semantic associations. This skill has several real-world applications, especially in the medical field where it can help with diagnosis.

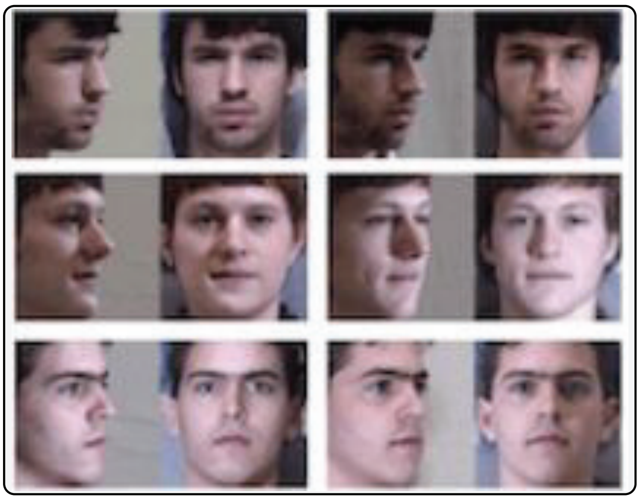


Fig. 26. Pictures showing how a GAN-based face frontal shot is produced that go beyond face rotation angles (adapted from [112, Fig. 2]).

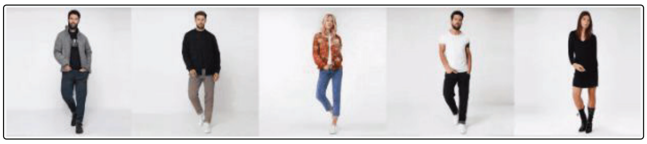


Fig. 27. Different human poses using GANs (adapted from [114, Fig. 1]).

H. Face Frontal View Generation

The report presented by R. Huang *et al.* discusses GAN for photo-realistic and identity-preserving frontal view synthesis, as illustrated in Fig. 26 [112], [113]. The work demonstrates how to use GANs to turn angle-shot images into frontal views of humans (see Fig. 26). The goal of the produced front-on photos is to input them into a system for face identification or verification.

I. Generating Novel Human Poses

In [114], a 2017 work by L. Ma *et al.* provides an example of how to generate new images of human models in new postures. As an illustration of this, Fig. 27 demonstrates how to use GANs to display different human positions in the pictures.

J. Images to Emojis

In 2016, Y. Taigman *et al.* used a GAN to convert images from a domain to a different one. For example, street numbers to MNIST handwritten digits and celebrity shots to what are called small cartoon faces or emojis. An illustration of this application is depicted in Fig. 28 [115].

K. Images Editing

In 2016, G. Perarnau *et al.* used a GAN model, named IcGAN, to reconstitute photos of faces with particular

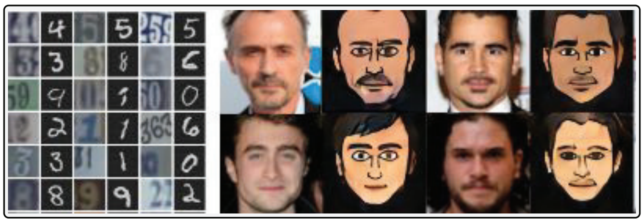


Fig. 28. SVHN to MNIST dataset and face images changed to emojis (adapted from [115, Fig. 9]).

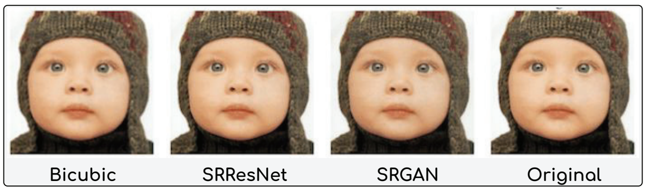


Fig. 29. GAN-generated images with super resolution (adapted from [18, Fig. 2]).

defined features, like variations in the color of the hair, style, expression of the face, and even gender [116]. Furthermore, MY. Liu *et al.* explored face generation having specific attributes such as glasses, facial expression, and hair color [117]. They also examined the creation of various visuals, such as scenes featuring a variety of hues and tones.

L. Face Ageing

2017 witnessed the release of a study showing that faces that appear to be between younger and older may be created using GANs [118]. A 2017 work by Z. Zhang on age progression/regression using conditional adversarial auto-encoder is another example involving de-aging facial portrait photographs using a GAN-based technique [119].

M. Photo Blending

Photo blending is the process of combining elements from two distinct images to produce a new one. For blending pictures, a 2017 paper by H. Wu *et al.* demonstrates how to mix images using GANs, especially images of large structures, fields, and mountains [120].

N. Super Resolution

In 2016, C. Ledig *et al.* demonstrated how to employ GANs, specifically the SRGAN model, to produce output images with greater, often much higher pixel resolution. This application is illustrated in Fig. 29 [18]. Furthermore, H. Bin *et al.* demonstrate that GANs can be used to generate versions of photographs of human faces [121].

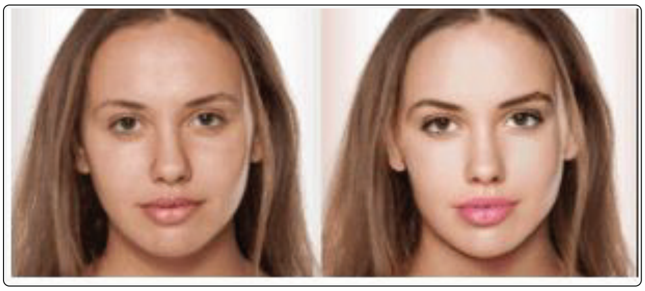


Fig. 30. Example of facial photos swapping (adapted from [65, Fig. 1]).

O. Photo In-painting

In a 2016 paper by D. Pathak *et al.* on context encoders for feature learning by inpainting, the authors considered how to fill in a portion of an image that was erased for any reason using GANs, namely context encoders, for photograph in-painting, also known as hole filling [122]. In [123] published in 2016, R. A. Yeh *et al.* use GANs to fill in and restore purposefully damaged pictures of human faces. Facebook's ExGAN, often known as ”eye-opening GAN,” is an example of this application.

P. Face Swapping

One of the most frequently mentioned applications of GANs is face swapping. Here, GANs can seamlessly swap the faces of two distinct individuals. Some of the most controversial face-swapping videos include celebrities' faces altered on a range of inappropriate outcomes. An example is illustrated in Fig. 30. The Open University of Israel and Bar-Ilan University created FSGAN. This variant of GAN is utilized for reenactment and face swapping. The fact that this method can be applied in real-time to any pair of people and does not require any prior training represents one of its key advantages. An enhanced version of face swapping is called deep fake, which involves swapping out a person's face in a video with another person. Among the most well-known applications that could show how face-swapping and deep fake work in practice are ZAO and the newly released FaceApp mobile application.

Q. Video Prediction

Having the ability to predict the future appears to be an unusual ability. But now that we have GANs, this looks normal. In this application, GANs enable the prediction of future frames in videos. Recently, the double-motion GAN architecture has learned to enforce the prediction of future frames following pixel-wise video flow. Furthermore, in 2016, C. Vondrick *et al.* explained how GANs are applied to video prediction. They successfully forecasted up to a second of video frames, mostly for static components of the scene [124].

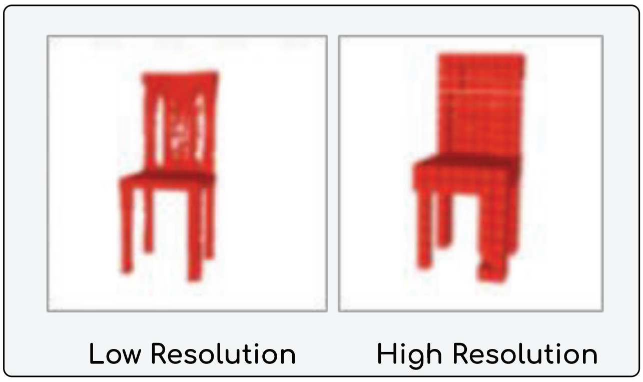


Fig. 31. Demonstration of GANs 3D low and high resolution imaging (adapted from [125, Fig. 3]).

R. Generating of 3D Objects

A study published in 2016 by J. Wu *et al.* describes a GAN for producing new 3D objects, like tables, chairs, couches, and cars (see Fig. 31 [125]).

S. Privacy-Preserving

Generative models can facilitate information sharing. This usually comprises detailed information, significant pattern-based data, and more shape-related data. Because of its architecture, which compels them to understand the difference between real and fake samples, GANs are being trained to identify harmful viruses and cyber-attacks. Secure Steganography Based on GANs (specifically SSGAN) is a technology that applies self-analysis to images to detect potentially dangerous encodings.

T. Domain Adaptation

Domain adaptation issues usually occur because a particular type of real-world data is used to train models. An example is the computer vision issue where the model fails due to variations in ambient lighting, camera angles, or atmospheric conditions. GANs help solve this problem by focusing on a specific section of an image to enhance the image resolution and quality; hence, producing better outcomes.

U. Drug Discovery

Performing research to identify the primary cause of an illness by examining the numerous protein types that could serve as the culprit to determine the primary cause of an illness, GAN-based models have demonstrated good performance and have contributed to a decrease in the amount of time and human involvement.

VII. Examples of GANs Real-World application Domain

After discussing theory and both academic and non-academic research, this section examines the applications of GANs in the commercial world. The practical use of GAN models appears to be only getting started, although research on the subject has been quite busy since 2014.

A. Medical Field

As a result of their capability to enhance images, GANs can be used to create photo-realistic super-resolution single images in medicine. In this field, GANs are in high demand due to the need for high-quality and precise images. Healthcare facilities require high-quality photographs. However, this can be difficult to obtain specific measurement protocols. Reducing radiation exposure for patients is imperative when employing low-dose computed tomography or magnetic resonance imaging scanning, as these procedures can be dangerous to those with certain underlying medical problems, like lung cancer and live tumors. The effect of the low-quality scans is that obtaining good-quality photographs becomes more difficult. Although super-resolution enhances collected images and effectively reduces noise, the medical field is slow to implement GANs since several trials and tests are required due to safety concerns. With the use of GAN technologies, numerous firms are devotedly striving to improve the lives of medical care providers though it suffers from some challenges like the instability mentioned above.

B. Marketing Industries

Businesses can forecast demand and modify their marketing plan by examining past data, market trends, and other pertinent variables. Marketing and sales professionals can determine price elasticity and set prices that optimize revenue and profitability by using demand forecasting.

Although artificial intelligence is not as effective at marketing as humans, humans become more proficient at work provided that they use AI technologies. AI can streamline daily operations to facilitate and expedite marketing procedures. Marketing routines are boosted by the use of GAN models like data segmentation, and customer journey mapping, to name a few. In particular, GAN models in advertising can save advertisers a great deal of pressure by streamlining the programmatic advertising process. Using the latest developments in GANs, Rosebud AI created generative photo applications that generate custom images of humans, for instance, fashion models, that do not exist. Also, they created the Tokkingheads application able to animate any facial image, whether synthetic or real, using text or audio as input. In the 3D graphics industries, synthetic faces proved to be advantageous, where the 2D face images are converted into 3D samples and used as assets for video games. The popularity of GANs in the mobile application market is a form of entertainment tool. FaceApp and ZAO applications are examples of mobile applications in this sector. These applications offer unique features such as the ability to alter someone's facial look or replace a celebrity face in a video with their faces. All these examples demonstrate that industries with the highest rate of GAN usage are found in the marketing and advertising sectors.

C. GANs As A Service

Some businesses provide access to GANs along with all the necessary interfaces and infrastructure to manage the data, train the models, and provide the desired outcomes. Among these businesses are runaway AI, content creators, prompting other interested parties to benefit from what the business refers to as ”generative media features,” which help bring GAN capabilities to a wider audience. However, for non-programmers, the use of GANs may be challenging without a graphical user interface.

D. GANs Technology For Dataset Generation in Computer Vision

GAN models can be used as a method to synthesize fresh labeled data samples from a comparatively limited amount of hand-crafted assets. This strategy appears to be quite promising, and in the future, more businesses will likely utilize it. The synthetic data offers considerably more precise annotations than the human ones could ever hope to provide, while also opening up a whole new world of possibilities in recreating extremely complex objects and settings. The generated data provides full control over the data variance (for example, we may choose the races, body forms, sizes, and proportions that we want to see in human body models).

E. Social Media

Chatbots are at the crossroads of AI, marketing, and customer support. Many chatbot services are now available on numerous platforms and from various brands as a result of the recent expansion of generative AI. The most notable are those on search engines such as Google Bard and Microsoft Copilot (Bing Chat). Social media sites are also involved, thanks to My AI for Snapchat and Meta AI on Instagram and Messenger. To customize user experiences, Meta AI offers a variety of personalities. Making suggestions to customers about what they should purchase will help them feel more understood. Thus, the development of Chatbots helps and facilitates the market sector growth rate.

VIII. Evaluation Metrics for the GANs' Performance

The performance evaluation process is a challenge associated with training GANs, which involves figuring out how well the model fits the data distribution. Theoretical and practical developments have greatly advanced, and a wide variety of GAN versions are currently accessible [126]. However, there has not been much work done to evaluate GANs, which has left certain gaps in evaluation techniques. The following section elaborates on some important and typical evaluation metrics for measuring the performance of GANs.

A. Visual Inspection

Visual inspection is one of the easiest and most natural ways to assess GAN performance. This entails assessing the generated samples' realism, diversity, and consistency by comparing them with the real data. Common GAN faults and defects, like mode collapse, blurriness, or distortion, can also be seen by visual inspection. However, because visual inspection relies on human perception and choice, it is arbitrary, inconsistent, and challenging to quantify. As such, it is not a reliable or objective way to compare different GAN models or settings.

B. Quantitative Metrics

Evaluating GAN quantitative performance can be done through metrics such as: (*i*) FID: It describes Wassertein-2 distance between the multivariate Gaussian that adapts the data embedded within the feature space [127]; (*ii*) IS: This metric is used to evaluate the quality of images generated by GANs. The process involves computing the mean Kullback-Leibler divergence between the conditional label distribution obtained from all samples and the marginal label distribution [127]; (*iii*) The maximum mean discrepancy (MMD). This distance lies in the probability measure space. It exploits the variations obtained from the probability of each sample distribution and can be used for two-sample testing; (*iv*) The kernel inception distance (KID). It calculates the difference between two distributions of probability using samples taken independently from each probability distribution; (*v*) The peak signal-to-noise ratio (PSNR). This metric measures how much two images' peak signal-to-noise ratios differ from one another, then compares the outcome to the matching real images [127]; (*vi*) The mode score (MS). Its role is to reflect the visual quality and variety of the generated image; *vii*) The structural similarity index measure (SSIM): It is used to forecast the degree of similarity between two images; (*viii*) The precision and recall (PR). The PR metric determines the percentage of generated samples that closely resemble real data and the percentage of real data that are covered by the generated samples; *(ix)* The learned perceptual image patch similarity (LPIPS). It evaluates the distance between image patches. Image patches with a low LPIPS score are more perceptually similar, while those with a higher LPIPS value are more perceptually dissimilar; *(x)* The error relative global dimensionless synthesis (ERGAS). ERGAS assesses the degree to which fused images adhere to two characteristics from both a monospectral and multispectral perspective; *(xi)* The blind image spatial quality evaluator (BRISQUE). This metric evaluates the quality of images with the same type of distortion; *(xii)* The root mean square error (RMSE). RMSE calculates the square root of the average differences between the actual and predicted values; (*xiii*) Diversity score (DS). This metric evaluates the pixel-wise variation diversity of the generated images. It is based on the idea that a high entropy GAN should generate unpredictable and non-repetitive images. DS is calculated by taking the average pairwise distance between the generated images, normalized by the average pairwise distance between the real images. A higher DS indicates greater picture diversity. However, it also has some limitations, such as being sensitive to the image resolution and not accounting for the semantic diversity of the images; (*xiv*) The perceptual path length (PPL). PPL evaluates the seamlessness and organic flow of the transition between the generated images. It is founded on the idea that a well-performing GAN should produce images that lie on a low-dimensional manifold, which allows for the smooth transformation of one image into another by changing the latent variables. It is calculated by normalizing the average perceptual distance between the interpolated and original images by the latent distance. A lower PPL results in smoother, more natural-looking images, but also has disadvantages, such as being computationally expensive and failing to consider the global structure of the manifold.

Quantitative measurements can provide more objective and consistent results than visual inspection, though they might not capture all aspects of GAN performance, such as semantic coherence or stylistic consistency. Additionally, because they rely on predefined criteria or pre-trained models, they can not align with human perception or choice. Furthermore, they may be sensitive to hyper-parameters, datasets, or reference models and may not generalize well across multiple tasks or domains.

C. User Studies

This method provides more accurate and direct feedback on how humans perceive and appreciate the generated samples. It can capture aspects not considered by visual inspection or quantitative metrics like semantic coherence, stylistic consistency, and domain relevance, to name a few. However, because they require recruiting, training, and compensating participants, User Studies can be expensive, time-consuming, and labor-intensive. They may also suffer from biases, noise, or inconsistencies resulting from participant background, knowledge, motivation, or mood. Furthermore, the outcome can also be impacted by the User studies' layout, style, or presentation. For example, the kind, scale, or language of the questions or ratings, or the quantity, arrangement, or size of the models or samples.

Mostly, there is no general agreement on the way of evaluating a given GAN-generated model. This is a problem for many researchers, especially when comparing different GAN model architectures, and model configurations or choosing a final GAN generator model during the training process. Furthermore, it should be noted that there are no objective evaluations of GANs generator models which remains an open problem in the research domain.

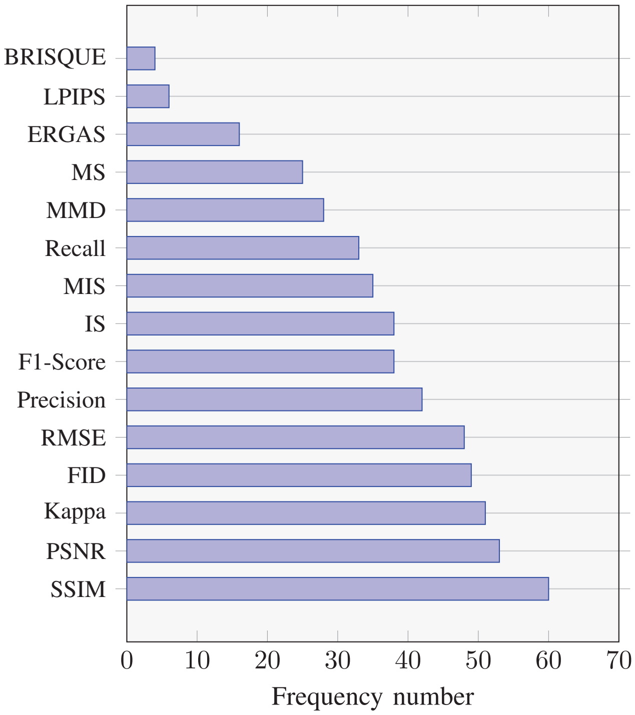


Fig. 32. GANs evaluation metrics. Note that we have listed only 15 metrics that we believe are the most significant ones.

D. GAN Evaluation Metric Frequency

Figure 32 depicts the frequency of different GANs performance evaluation metrics. This provides an idea of which performance metrics are best to be used for GAN evaluations. For example, SSIM is the most used followed by PSNR, Kappa, and FID, while BRISQUE is the less-used metric.

IX. Meta-Analysis of GANs on Different Applications

This section summarizes the evolution of GANs and related publications.

A. Evolution of GAN Model

As seen in Table I, the evolution and development of GANs increased from 2014 to 2020 with different architecture and the loss function due to the growing advantages of GANs in various applications. In 2014, GANs were introduced in their simplest form (Vanilla GAN). During the same year, GAN innovation led to the creation of cGAN. In 2015, LAPGAN with the same architecture as Vanilla GAN was introduced, then followed by adversarial autoencoders

TABLE I
BRIEF HISTORY OF GANS MODELS FROM 2014 TO 2020.

	2014	2015	2016	2017	2018	2019	2020
Architecture-variant	1. *Vanilla GAN*: Original GAN. 2. *cGAN*: Label info to D and G.	1. *LAPGAN*: Uses laplacian pyramid. 2. *Adversarial autoencoders GAN*: Uses NAS, network architecture GAN.	1. *DCGAN*: Used transpose convolutional layers. 2. *Text-to-image GAN*: Semantics features. 3. *InfoGAN*: Classifier Q for labdels, used regularization mode.	1. *CycleGAN*: Image style transfer. 2. *SRGAN*: Used residual blocks. 3. *ACGAN*: Auxilary classifer. 4. *Pix2Pix* GAN: Based on cGAN. 5. *DVD-GAN*: Based on BigGAN, uses 3D convolutions. 6. *StackGAN*: Used two stages. 7. *PGAN*: Progressive add layers when training	1. *StyleGAN*: Scale specific face generation.	1. *BigGAN*: Self-attention module, larger batch size, hinge loss. 2. *StackGAN++*: Multiple generators and multiple generator.	
Loss-variant				1. *LSGAN*: Personal divergence, 2. *WGAN*: Wasserstein distance, 3. *EBGAN*: Based on deep convolutional architecture		1. *DiscoGAN*: Cross domain relations	1. *MRGAN*: Mode regularized

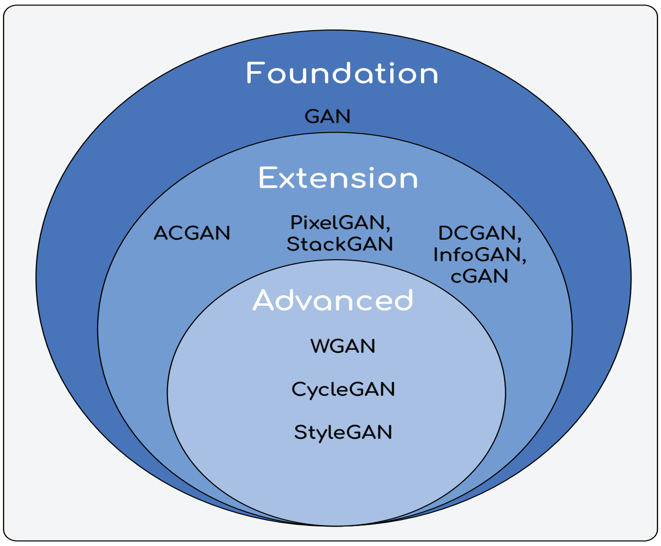


Fig. 33. GANs variants architecture development and evolution.

GAN with loss function change. In 2016, DCGAN, text-to-image GAN was created with the same design and then a few months later, InfoGAN was introduced with a loss function change. As GAN architecture continues to evolve year after year, new versions of GANs have been created with different designs, functionalities, and applications. From 2014 to the present, GAN applications have increased and are still increasing, bringing solutions to unsupervised learning problems as illustrated in Fig. 33.

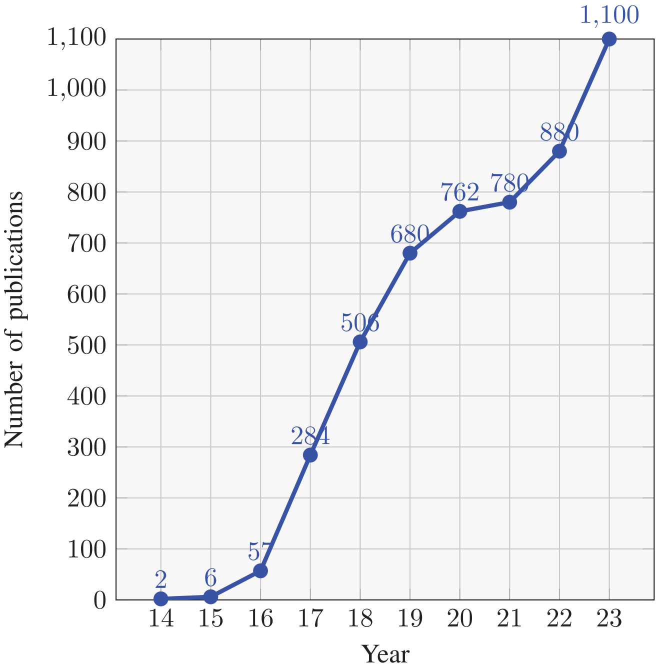


Fig. 34. Approximated cumulative number of GAN-related papers published from 2014 to 2023.

B. A Summary of Publications on GANs from 2014 to 2023

In addition to the evolution of GANs from 2014 to 2023, Fig. 34 depicts a summary of GAN-based published papers during the same period. We learn from this figure that extensive work has been done between 2014 and 2019, which resulted in the advancement in GANs and

TABLE II
TIMELINE OF SOME GANS PUBLICATIONS FROM 2014 TO 2024.

No.	Date	Reference	Description	Comments
1	Nov. 2014	[1]	Generative adversarial networks	The authors explain generative models goals and training probabilities using GANs as examples.
2	Jun 2016	[31]	Improved techniques for training GANs	The paper presents multiple new architectural features and training procedures for GANs' framework.
3	Jun. 2017	[5]	Using fader networks to manipulate images by sliding attributes	This paper introduces a new encoder-decoder architecture trained to reconstruct images.
4	May 2017	[19]	Using deep convolutional neural networks to classify ImageNet	In this paper, the algorithm trains a large, deep convolutional neural network for classifying the 1.3 million high-resolution images.
5	Sep. 2017	[23]	An introduction to GANs and their outlook	The paper summarizes the GANs state-of-the-art considering the future.
6	Nov. 2018	[34]	Inverting the GAN's generator	The authors introduce a technique to project image samples to the latent space utilizing a pre-trained GAN.
7	Oct. 2019	[35]	Image generation with GANs-based techniques: A survey	The paper discusses five applicable and fascinating areas for image synthesis based on GANs techniques.
8	Jan. 2019	[61]	GAN generators with a style-based approach	The authors propose new automated methods applicable to any generator architecture.
9	Jun. 2019	[48]	Image de-raining based on a conditional GAN	The authors propose a method denominated image de-raining conditional GAN for performance evaluations.
10	Jan. 2019	[107]	Evolutionary GANs	The paper presents a new GAN framework that enhances GAN stability during training and boosts overall performance.
11	Mar. 2020	[25]	A review of GANs and their application in cybersecurity	in this paper, GANs are discussed in terms of their strengths, their weaknesses, and the pros and cons of other models.
12	Jul. 2021	[26]	A comprehensive survey and evaluation of networking using GANs	The authors explore how GANs can be beneficial to various aspects of communication systems and computer networks.
13	May 2022	[46]	Loss functions of GANs: Opportunities and challenges	The authors conduct a survey of GAN loss functions and analyze their advantages and disadvantages
14	Dec. 2023	[106]	Various applications of GANs in various disciplines, ranging from medicine to remote sensing	The paper describes GAN applications across a wide range of fields, including medicine and remote sensing.
15	Mar. 2024	[86]	The authors improve GANs for the generation of color images of faces	The authors develop an improved GAN that is specifically designed to generate color portraits from sketches.

AI observed today. This also demonstrates that due to the growing usage of GANs in various applications, a growing number of papers have been published. The first two GAN journal articles were published in 2014 [1], [62]. There were about 57 publications on GANs in 2016, while in 2017, more than 200 papers were published. In 2023, further relevant papers were released, and more than 1000 papers on GANs underwent review. Table II shows some examples of publications on GANs and their descriptions.

C. Cumulative Number of GAN Applications

Figure 35 depicts several studies on GAN applications. As seen from the chart, numerous research studies are related to GANs in computer science (examples are computer vision, sequential data, image synthesis and manipulation, image super-resolution) (application area number = 3950), Engineering (communications, robotics, control systems) (application area number = 1571), mathematics (multivariate calculus, linear algebra, non-linear programming, basic statistics) (application area number = 1114), physics (astronomy) (application area number = 449), health (medical) (application area number = 269) and other application domains (such as social science, art and humanities) (application area number = 1775), respectively. Figure 35 also shows several studies on GAN applications. As seen from the chart, numerous research studies are based on GANs and target domains like:

- computer science: computer vision, sequential data, image synthesis and manipulation, image super-resolution, application area number = 3950;

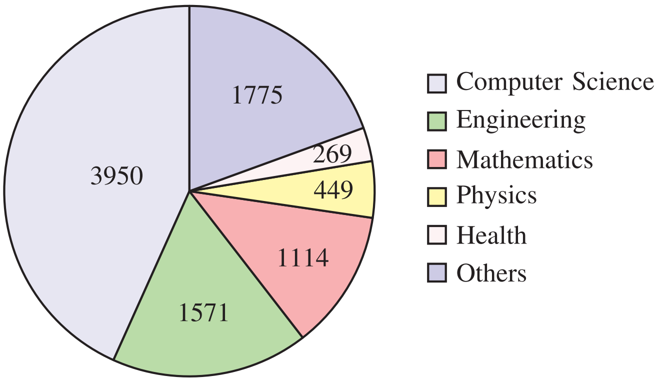


Fig. 35. Approximated cumulative number of GAN applications reviewed. Note that we highlight the most used application domains benefiting from GAN models.

- engineering: communications, robotics, control systems, application area number = 1571;
- mathematics: multivariate calculus, linear algebra, nonlinear programming, basic statistics, application area number = 1114;
- physics: astronomy, application area number = 449;
- health: medical, application area number = 269;
- Other domains: social sciences, arts and humanities sciences, earth and planetary sciences, decision sciences, materials science, application area number = 1775.

The growing number of published papers is depicted in Fig. 34 and the charts provided in Figs 32 and 35 justify the ever-growing number of GAN variants. Table III summarizes some of the most prominent GAN variants. The table also shows that each of these variants has several advantages and drawbacks. Given that they are designed for specific applications, techniques to mitigate these impediments are introduced.

X. Open Challenges and Way Forward

All of the GAN architectures that have been discussed in this review share one feature. Each is based on the adversarial loss principle, and all have generators and discriminators that exploit each other's adversarial characteristics. Over the past few years, GANs have demonstrated remarkable progress and have become one of the most discussed topics in the field of ML. It is believed that GANs have a promising future because research is being conducted to overcome their shortcomings and difficulties. It is anticipated that GANs will be used in many more fascinating applications in a variety of sectors, including e-commerce, agriculture, and entertainment. Furthermore, improvements in GANs might lead to the creation of more complex technologies, such as deep fake technologies, face swapping, and computer vision, which may have a significant impact on society.

Deep fakes might be entertaining, but they also have some drawbacks. For example, there is a high potential for disinformation to proliferate and privacy issues. The development of techniques to identify and counteract the harmful impacts of deep fake and other malicious uses of GAN-generated content will become increasingly important as GANs get more sophisticated and represent a significant area of open research.

Morph, produced by landmark-based or GAN-based techniques, comprises two unique facial images paired with biometric face recognition system information about two people's identities. In the literature, Morph is seen as a challenging attack for face recognition systems. This requires research to differentiate between favorable and adverse effects.

Furthermore, GANs have been reported to generate very realistic anime characters with some sophisticated architectures. In a few years, we might be able to generate more technologies using AI. With ongoing research aimed at addressing their challenges and limitations, it is legitimate to be optimistic about GAN's future despite its challenges. As GANs continue to improve, there should be more creative uses across a range of sectors. In addition, we should anticipate the new opportunities and problems brought about by these developments. For example, the creation of a Police GAN to counter disinformation will represent an advancement in GANs. This may not be an easy task and is an area for further investigation.

Currently, there is no established consensus on ethical AI policies. Consequently, an ethical GAN has not yet benefited from an international framework that GAN developers and users should follow. Given the evolution happening in this field, governments and legislatures must propose an international framework that binds all GAN developers and users globally. Proposing a framework or policies for a moral GAN is a crucial field of study and an idea for further development. This framework will include AI regulations and policies for AI-generated content monitoring. The framework will provide guidance for the development, deployment, and use of AI technologies, such as GANs. Some significant points to be included and emphasized are the respect of human rights, fairness promotion and non-discrimination. The framework should also consider privacy and data protection, transparency and explainability, safety and security, and be human-centered. The policies should also address environmental and social sustainability. This represents the way forward.

XI. Conclusion

The ability of GANs to produce data identical to the original data has revolutionized many industries, including creativity. In this paper, a general review of GANs has been presented to explain the reasoning behind their creation. We examined the foundations, background, and range of GAN variations, each designed for a specific purpose. In our study, we have found that GANs improve AI capabilities. It is anticipated that GANs ar expected to transform AI and provide new opportunities as they develop further. This

TABLE III
SUMMARY ON GANS VARIANTS, APPLICATIONS, ADVANTAGES, CHALLENGES, AND COMMENTS [41].

No.	GAN Variant	Application	Advantages	Challenges	Comments
1	Vanilla GAN	Text-to-image conversion	Realistic image, improved accuracy	Mode collapse, low performance	Use multi-layer perceptrons and stochastic gradient descent. Also required enhancement.
2	cGAN	Image synthesis and manipulation	Good quality images	Much data required	Use class labelled to generate specified target convergence faster.
3	DCGAN	Image processing and natural language processing	Stable, high resolution, and good quality images	Layers are not fully connected	Use convolutional and convolutional transpose layers.
4	StackGAN, StackGAN++	Text-to-image conversion	Generate high-fidelity images	Use conditional augmentation and residual blocks	Use two different stages.
5	WGAN	Image synthesis and manipulation	Stable, high quality images	Computational expensive	Use gradient descent in the training process.
6	InfoGAN	Image synthesis and manipulation	Good for optimazation	Unstable, mode collapse	Use regularization methods.
7	LSGAN	Image translation	Stable, high quality images	Reduce sample diversity	Use objective function for training.
8	ACGAN	Medical imaging	Stable, high quality images	Mode collapse	Incorporates class information into GAN architecture.
9	DVD-GAN	Video processing	Low computation cost	Memory constraint, computational expensive	Use two generators.
10	PGAN	Image translation	High resolution images	Mode collapse, unstable	Use more layers to maximize the training process.
11	Text-to-image GAN	Natural language processing	Good quality images	Computational and memory expensive	Facilitate text-to-image synthesis.
12	StyleGAN, StyleGAN2, StyleGAN2-ADA, StackGAN3	Image translation	High resolution images and good quality images	Computational expensive	Use PGAN and neural style transfer technology.
13	CycleGAN	Image translation	Simple and stable	Data availability and quality	Image-to-image translation without database training.
4	DiscoGAN	Image translation	Simple, accurate and low loss function	Memory constraint	Identify cross domains relationship from data.
15	Pix2Pix GAN	Image-to-image conversion	Effective at learning conditional distribution	Require specific models and loss functions	Ease control on image generated.
16	BigGAN	Image resolution	High quality, realistic images and high performance using Frecher inception distance	Memory constraint, computational expensive	Produce large-scale images and increase the image capabilities.
17	SRGAN	Computer vision	High resolution	Insufficient feature information extraction ability, mode collapse, unstable	Use deep neural network, upscaling low resolution to high-resolution images.
18	LAPGAN	Image synthesis and manipulation	High-quality images	Computational expensive	Use multiple numbers of generator and discriminator.
19	Adversarial autoencoder GAN	Videos and 3D images creation	Very effective	computationally expensive, optimization problems, ethical problems, social implications	Use regularizer for the latent space.
20	EBGAN	Infusion detection and prediction	High performance, stable	Unstable, sensitive data	Use autoencoders and regularizer for solving evaluation metrics problems using energy.
21	MRGAN	Image generation	Diverse, stable training process, and high-quality images	Mode collapse	Generate semantically consistent images by effectively capturing visual textual consistency.

study also demonstrated several GAN applications and discussed their disadvantages. Concerning the risks associated with the development of GANs, it is obvious that further research is required to counteract the dangers associated with their incorrect use. Therefore, the way forward has been discussed. We can legitimately claim that GAN is a revolution despite these risks.

References

[1] I. Goodfellow *et al.*, "Generative adversarial networks," *Adv. in Neural Inf. Process. Syst.*, vol. 27, p. 2672–2680, Nov. 2014.

[2] E. Brophy *et al.*, "Quick and easy time series generation with established image-based GANs," *ArXiv*, p. 2672–2680, Feb. 2019.

[3] KG. Hartmann *et al.*, "EEG-GAN: Generative adversarial networks for electroencephalograhic (EEG) brain signals," *ArXiv*, Jun. 2018.

[4] TR. Shaham *et al.*, "SinGAN: Learning a generative model from a single natural image, Seoul, Korea (South)," in *Proc. IEEE/CVF Int. Conf. on Comput. Vision (ICCV), Seoul, Korea (South)*, 27 Oct.-02 Nov. 2019, pp. 4570–4580.

[5] G. Lample *et al.*, "Fader networks: Manipulating images by sliding attributes," *Adv. in Neural Inf. Process. Syst.*, vol. 30, p. 5963–5972, Jun. 2017.

[6] R. Gal *et al.*, "StyleGAN-NADA: CLIP-guided domain adaptation of image generators," *ACM*, vol. 41, pp. 1–3, Jul. 2022.

[7] R. Abdal *et al.*, "Tyleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows," *ACM*, vol. 40, pp. 1–21, May 2021.

[8] W. Xia *et al.*, "Gan inversion: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, pp. 38–3121, Jun. 2022.

[9] A. Ramesh *et al.*, "Zero-shot text-to-image generation," in *Proc. of the 38th Int. Conf. on Mach. Intell., Vitual*, vol. 139, 18-24 Jul. 2021, pp. 8821–8831.

[10] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. 38th Int. Conf. Mach. Intell.*, vol. 139, 18-24 Jul. 2021, pp. 8748–8763.

[11] O. Patashnik *et al.*, "Text-driven manipulation of StyleGAN imagery," in *Proc. IEEE/CVF Int. Conf. on Comput. Vision, Montreal, QC, Canada*, 10-17 Oct. 2021, pp. 2085–2094.

[12] W. Fedus *et al.*, "MaskGAN: Better text generation via filling in the." *ArXiv*, vol. 1, Jan. 2023.

[13] N. Jetchev *et al.*, "Texture synthesis with spatial generative adversarial networks," *ArXiv*, Nov. 2016.

[14] J. Guo *et al.*, "Long text generation via adversarial training with leaked information," in *Proc. AAAI Conf. Comput. Intell., Washington DC, USA*, vol. 32, no. 1, 27 April. 2018.

[15] GK. Dziugaite *et al.*, "Training generative Neural networks via maximum mean discrepancy optimization," *ArXiv*, May 2015.

[16] T. Karras *et al.*, "Progressive growing of GANs for improved quality, stability, and variation," *ArXiv*, Oct. 2017.

[17] SA. Hussein *et al.*, "Image-adaptive GAN based reconstruction," *ArXiv*, Jun. 2019.

[18] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR), Honolulu, HI, USA*, 21-26 Jul. 2017, p. 105–114.

[19] J. Xu *et al.*, "An overview of deep generative models," *IETE Technical Review*, vol. 32, no. 2, pp. 131–139, 2015.

[20] V. Mannam *et al.*, "Performance analysis of semi-supervised learning in the small-data regime using vaes," *arXiv*, 2020.

[21] H. Shao *et al.*, "Controlvae: Tuning, analytical properties, and performance analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 12, pp. 9285–9297, 2021.

[22] C. Leke *et al.*, "Modeling of missing data prediction: Computational intelligence and optimization algorithms," in *Proc. IEEE Int. Conf. Sys., Man, and Cybernetics (SMC)*, 05-08 October 2014, pp. 1400–1404.

[23] ——, "Deep learning-bat high-dimensional missing data estimator," in *Proc. IEEE Int. Conf. Sys., Man, and Cybernetics (SMC)*, 05-08 October 2017, pp. 483–488.

[24] DO. Esan *et al.*, *Generative adversarial networks: Applications, challenges, and open issues*, 1st ed. IntechOpen, 2023.

[25] W. He *et al.*, "A survey on uncertainty quantification methods for deep learning," *arXiv*, 2023.

[26] E. Daxberger *et al.*, "Mixed-variable bayesian optimization," *arXiv*, 2019.

[27] A. Krizhevsky *et al.*, "ImageNet classification with deep convolutional Neural networks," *Adv. in Neural Info. Process. Syst.*, vol. 60, no. 6, pp. 84–90, May 2017.

[28] M. Mohebbi Moghaddam *et al.*, "Games of gans: Game-theoretical models for generative adversarial networks," *Artificial Intelligence Review*, vol. 56, no. 9, pp. 9771–9807, 2023.

[29] X. Guo *et al.*, "Gans training: A game and stochastic control approach," *Mathematical Finance*, vol. 34, no. 2, pp. 522–556, 2024.

[30] Z. Ahmad *et al.*, "Understanding gans: Fundamentals, variants, training challenges, applications, and open problems," *Multimedia Tools and Applications*, pp. 1–77, 2024.

[31] T. Chakraborty *et al.*, "Ten years of generative adversarial nets (GANs): a survey of the state-of-the-art," *Mach. Learning: Sci. and Technol.*, vol. 5, no. 1, p. 25, Jan. 2024.

[32] Z. Pan *et al.*, "Recent progress on generative adversarial networks (GANs): A survey," *IEEE access*, vol. 7, pp. 36 322–36 333, Mar. 2019.

[33] H. Alqahtani *et al.*, "Applications of generative adversarial networks (GANs): An updated review," *Archives of Comput. Methods in Eng.*, vol. 28, pp. 52–552, Mar. 2021.

[34] K. Wang *et al.*, "Generative adversarial networks: Introduction and outlook," *IEEE/CAA J. of Automatica Sinica*, vol. 4, no. 4, pp. 588–598, Sep. 2017.

[35] B. Yilmaz *et al.*, "Understanding the mathematical background of Generative Adversarial Networks (GANs)," *Mathematical Modelling and Numerical Simulation with Appl.*, vol. 3, no. 3, pp. 234–255, Sep. 2023.

[36] C. Yinka-Banjo *et al.*, "A review of generative adversarial networks and its application in cybersecurity," *AI. Review*, vol. 53, pp. 1721–1736, Mar. 2020.

[37] H. Navidan *et al.*, "Generative adversarial networks (GANs) in networking: A comprehensive survey & evaluation," *Comput. Netw.*, vol. 194, pp. 108–149, Jul. 2021.

[38] H. Lin *et al.*, "How generative adversarial networks promote the development of intelligent transportation systems: A survey," *IEEE/CAA J. of Automatica Sinica*, Aug. 2023.

[39] K. Liu *et al.*, "FISS GAN: A generative adversarial network for foggy image semantic segmentation," *IEEE/CAA J. of Automatica Sinica*, vol. 8, no. 8, pp. 1428–1439, Jun. 2021.

[40] J. Gui *et al.*, "A review on generative adversarial networks: Algorithms, theory, and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 35, pp. 3313–3332, 23 Nov. 2021.

[41] D. Saxena *et al.*, "Generative adversarial networks (GANs) challenges, solutions, and future directions," *ACM Comput. Surveys (CSUR)*, vol. 54, no. 3, pp. 1–42, May 2021.

[42] Y. Hong *et al.*, "How generative adversarial networks and their variants work: An overview," *ACM Comput. Surveys (CSUR)*, vol. 52, no. 1, pp. 1–43, Feb. 2019.

[43] J. Abdul *et al.*, "A survey on generative adversarial networks: Variants, applications, and training," *ACM Comput. Surveys (CSUR)*, vol. 54, no. 8, pp. 1–49, Oct. 2021.

[44] T. Salimans *et al.*, "Improved techniques for training GANs," *Adv. in Neural Inf. Process. Syst.*, p. 29, Jun. 2016.

[45] T. Hinz *et al.*, "Improved techniques for training single-image GANs," in *Proc. IEEE Winter Conf. on Appl. of Comput. Vision (WACV), Waikoloa, HI, USA*, 03-08 Jan. 2021, pp. 1300–1309.

[46] I. Gulrajani *et al.*, "Improved training of wasserstein GANs," *Adv. in Neural Inf. Process. Syst.*, vol. 30, 2017.

[47] A. Creswell *et al.*, "Inverting the generator of a generative adversarial network," *IEEE Trans. on Neural Netw. and Learning Syst.*, Nov. 2018.

[48] L. Gonog *et al.*, "A review: generative adversarial networks," in *2019 14th IEEE conference on industrial electronics and applications (ICIEA)*. IEEE, 19 Jun. 2019, pp. 505–510.

[49] SN. Esfahani *et al.*, "Image generation with GANs-based techniques: A survey," *AIRCC's Int. J. of Comput. Sci. and Inf. Technol.*, vol. 1, pp. 33–50, Oct. 2019.

[50] GH. De Rosa *et al.*, "A survey on text generation using generative adversarial networks," *Pattern Recognition*, vol. 119, pp. 98–108, Nov. 2021.

[51] MJ. Kusner *et al.*, "GANs for sequences of discrete elements with the gumbel-softmax distribution," *ArXiv*, Nov. 2016.

[52] CJ. Maddison *et al.*, "The concrete distribution: A continuous relaxation of discrete random variables," *ArXiv*, Nov. 2016.

[53] RJ. Williams *et al.*, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learning*, vol. 8, pp. 229–256, May 1992.

[54] J. Guo *et al.*, "Long text generation via adversarial training with leaked information," in *Proc. of the AAAI Conf. on AI*, vol. 32, no. 1, 2 Feb. 2018, p. 5141–5148.

[55] L. YU *et al.*, "SeqGAN: Sequence generative adversarial nets with policy gradient," in *Proc. of the AAAI Conf. on AI*, vol. 31, no. 1, 13 Feb. 2017.

[56] K. Lin *et al.*, "Adversarial ranking for language generation," *Adv. in Neural Inf. Process. Syst.*, vol. 30, 2017.

[57] R. Dey *et al.*, "RankGAN: A maximum margin ranking gan for generating faces," *ArXiv*, Dec. 2018.

[58] CH. Lee *et al.*, "MaskGAN: Towards diverse and interactive facial image manipulation," in *Proc. of the IEEE/CVF Conf. on Comput. Vision and Pattern Recognition*, 13-19 Jun. 2020, pp. 5549–5558.

[59] M. Durgadevi *et al.*, "Generative adversarial network (GAN): A general review on different variants of GAN and applications," in *Proc. IEEE 6th Int. Conf. on Commun. and Electron. Syst. (ICCES), Coimbatre, India*, 08-10 Jul. 2021.

[60] Z. Zhao *et al.*, "Image augmentations for GAN training," *ArXiv*, Jun. 2020.

[61] Z. Pan *et al.*, "Loss functions of generative adversarial networks (GANs): Opportunities and challenges," *IEEE Trans. on Emerging Topics in Comput. Intell.*, vol. 4, no. 4, pp. 500–522, May 2020.

[62] M. Mirza *et al.*, "Conditional generative adversarial nets," *ArXiv*, Nov. 2014.

[63] H. Zhang *et al.*, "Image de-raining using a conditional generative adversarial network," *IEEE Trans. on Circuits and Syst. for Video Technol.*, vol. 30, no. 11, pp. 3943–3956, Jun. 2019.

[64] G. Antipov *et al.*, "Face aging with conditional generative adversarial networks," in *Proc. IEEE Int. Conf. on Image Process., Beijing, China*, 17-20 Sep. 2017.

[65] Z. Wang *et al.*, "Generative adversarial networks in computer vision: A survey and taxonomy," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.

[66] XL. Hou *et al.*, "Conditional GANs with auxiliary discriminative classifier," in *Proc. Int. Conf. on Mach. Learning*, 28 Jun. 2022, pp. 8888–8902.

[67] A. Odena *et al.*, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. on Mach. Learning*, 17 Jul. 2017, pp. 2642–2651.

[68] J. Cho *et al.*, "Conditional activation GAN: Improved auxiliary classifier GAN," *IEEE Access*, vol. 8, Dec. 2020.

[69] C. Bodnar *et al.*, "Text to image synthesis using generative adversarial networks," *arXiv*, May 2018.

[70] H. Zhang *et al.*, "StackGAN++: Realistic image synthesis with stacked generative adversarial networks," *IEEE Access*, vol. 41, pp. 1947–962, Jul. 2018.

[71] SK. Alhabeeb *et al.*, "Text-to-image synthesis with generative models: Methods, datasets, performance metrics, challenges, and future direction," *IEEE Access*, vol. 12, pp. 24 412–24 427, Feb. 2024.

[72] K. Dhivya *et al.*, "Text to realistic image generation using stackGAN," in *Proc. of the 7th Int. Conf. on Smart Structures and Syst.(ICSSS), Chennai, India*, 23-24 Jul. 2020, pp. 1–7.

[73] M. Arjovsky *et al.*, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. on Mach. Learning*, 17 Jul. 2017, pp. 214–223.

[74] A. Brock *et al.*, "Large scale GAN training for high fidelity natural image synthesis," *ArXiv*, Sep. 2018.

[75] T. Miyato *et al.*, "Spectral normalization for generative adversarial networks," *ArXiv*, Feb. 2018.

[76] X. Chen *et al.*, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," *Adv. in Neural Inf. Process. Syst.*, vol. 29, 2016.

[77] Y. Wang *et al.*, "InfoGAN: Improved information maximizing generative adversarial networks," in *Proc. IEEE 5th Int. Conf. on Mechanical, Control and Comput. Eng.,Harbin, China*, 25-27 Dec. 2020.

[78] X. Mao *et al.*, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 2794–2802.

[79] A. Clark *et al.*, "Adversarial video generation on complex datasets," *ArXiv*, Jul. 2019.

[80] T. Kim *et al.*, "Learning to discover cross-domain relations with generative adversarial networks," in *Proc. Int. Conf. on Mach. Learning*, 17 Jul. 2017, pp. 1857–1865.

[81] T. Karras *et al.*, "A style-based generator architecture for generative adversarial networks," *IEEE Trans. on Pattern Anal. and Mach. Intell.*, vol. 43, pp. 4401–4410, Jan. 2019.

[82] J. Zhu *et al.*, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 2223–2232.

[83] A. Almahairi *et al.*, "Augmented CycleGAN: Learning many-to-many mappings from unpaired data," in *Proc. IEEE Int. Conf. on Mach. Learning*, 3 Jul. 2018, pp. 195–204.

[84] T. Kaneko *et al.*, "Cyclegan-VC2: Improved CyclegGAN-based non-parallel voice conversion," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Process.,Brighton, UK*, 12-17 May 2019.

[85] T. Wang *et al.*, "Cyclegan with better cycles," *arXiv*, Aug. 2024.

[86] T. Angsarawanee *et al.*, "Generating images with desired properties using the DiscoGAN model enhanced with repeated property construction," in *Proc. 1st Int. Conf. on Adv. Inf. Sci. and Syst.*, 15 Nov. 2019, pp. 1–9.

[87] A. Wang *et al.*, "Application of generative adversarial network on image style transformation and image processing," Ph.D. dissertation, UCLA, 2018.

[88] P. Isola *et al.*, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition*, 2017, pp. 1125–1134.

[89] K. Lata *et al.*, "Image-to-image translation using generative adversarial network," in *Proc. IEEE 3rd Int. Conf. on Electron., Commun. and Aerospace Technol.*, 12-14 Jun. 2019, pp. 1125–1134.

[90] EL. Denton *et al.*, "Deep generative image models using a laplacian pyramid of adversarial networks," *Adv. in Neural Inf. Process. Syst.*, vol. 28, 2015.

[91] H. Yin *et al.*, "Laplacian pyramid generative adversarial network for infrared and visible image fusion," *IEEE Signal Process. Letters*, vol. 29, Sep. 2022.

[92] J. Zhao *et al.*, "Energy-based generative adversarial network," in *Proc. 5th Int. Conf. on Learning Representations, ICLR, France, Toulon*, 24-26 Apr. 2017, pp. 4681–4690.

[93] L. Ruthotto *et al.*, "An introduction to deep generative modeling," *GAMM-Mitteilungen*, vol. 44, no. 2, Jun. 2021.

[94] V. Kushwaha *et al.*, "Study of prevention of mode collapse in generative adversarial network (GAN)," in *Proc. IEEE 4th Conf. on Inf. Commun. Technol., Chennai, India*, 03-05 Dec. 2020.

[95] L. Metz *et al.*, "Unrolled generative adversarial networks," *ArXiv*, Nov. 2016.

[96] M. Soloveitchik *et al.*, "Conditional frechet inception distance," *ArXiv*, Mar. 2021.

[97] H. Mansourifar *et al.*, "Virtual big data for GAN based data augmentation," in *Proc. IEEE Int. Conf. on Big Data, Los Angeles, CA, USA*, 09-12 Dec. 2019, pp. 1478–1487.

[98] R. Adams *et al.*, *The new empire of AI: the future of global inequality*. John Wiley & Sons, Nov. 2024.

[99] NT. Tran *et al.*, "On data augmentation for GAN training," *IEEE Trans. on Image Process.*, vol. 30, pp. 1882–1897, Jan. 2021.

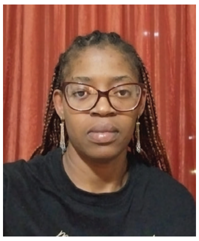
[100] S. Qian *et al.*, "Make a face: Towards arbitrary high fidelity face manipulation," in *Proc. IEEE/CVF Int. Conf. Comput. Vision, Seoul, Korea (South)*, 27 Oct.-02 Nov. 2019, pp. 33–42.

[101] C. Yu *et al.*, "Towards high-fidelity text-guided 3D face generation and manipulation using only images," in *Proc. IEEE/CVF Int. Conf. Comput. Vision, Paris, France*, 01-06 Oct. 2023, pp. 15 326–15 337.

[102] M. Brundage *et al.*, "The malicious use of artificial intelligence: Forecasting, prevention, and mitigation," *ArXiv*, Feb. 2018.

[103] D. Bazarkina *et al.*, "Malicious use of artificial intelligence," *Russia in Global Affairs*, vol. 18, no. 4, pp. 154–177, 2020.

[104] A. Annor *et al.*, "Application of artificial intelligence in forecasting: A systematic review," *ArXiv*, Nov. 2019.
[105] al. Khassawneh *et al.*, "A review of artificial intelligence in security and privacy: Research advances, applications, opportunities, and challenges," *Indonesian J. of Sci. and Tech.*, vol. 8, no. 1, pp. 79–96, Nov. 2023.
[106] C. Yeong-Hwa *et al.*, "Color face image generation with improved generative adversarial networks," *Electron.*, vol. 13, no. 7, Mar. 2024.
[107] Y. Jin *et al.*, "Towards the automatic anime characters creation with generative adversarial networks," *ArXiv*, vol. 92, pp. 1–6, Aug. 2017.
[108] P. Pradhyumna *et al.*, "A survey of modern deep Learning based generative adversarial networks (GANs)," in *Proc. 6th Int. Conf. Comput. methodologies and Commun. (ICCMC), Erode, India*, 29-31 Mar. 2022, pp. 1146–1152.
[109] P. Isola *et al.*, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 1125–1134.
[110] H. Zhang *et al.*, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. IEEE Conf. Comput. Vision (ICCV), Venice, Italy*, 22-29 Oct. 2018, pp. 5907–5915.
[111] T. Wang *et al.*, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition, Salt Lake City, UT, USA*, 18-23 Jun. 2018, pp. 8798–8807.
[112] R. Huang *et al.*, "Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis," in *Proc. IEEE Int. Conf. Comput. Vision (ICCV), Venice, Italy*, 22-29 Oct. 2017, pp. 2439–2448.
[113] Y. Li *et al.*, "Generative face completion," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR), Honolulu, HI, USA*, 21-26 Jul. 2017, pp. 3911–3919.
[114] L. Ma *et al.*, "Pose guided person image generation," *Adv. in Neural Inf. Process. Syst.*, vol. 30, 2017.
[115] Y. Taigman *et al.*, "Unsupervised cross-domain image generation," *ArXiv*, Nov. 2016.
[116] G. Perarnau *et al.*, "Invertible conditional GANs for image editing," *ArXiv*, Nov. 2016.
[117] MY. Liu *et al.*, "Coupled generative adversarial networks," *Adv. in Neural Inf. Process. Syst.*, vol. 29, 2016.
[118] G. Antipov *et al.*, "Face aging with conditional generative adversarial networks," in *Proc. IEEE Int. Conf. Image Process. (ICIP), Beijing, China*, 17-20 Sep. 2017, pp. 2089–2093.
[119] Z. Zhang *et al.*, "Age progression/regression by conditional adversarial autoencoder," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR), Honolulu, HI, USA*, 21-26 Jul. 2017, pp. 5810–5818.
[120] H. Wu *et al.*, "Gp-GAN: Towards realistic high-resolution image blending," in *Proc. IEEE 27th ACM Int. Conf. Multimedia*, 15 Oct. 2019, pp. 2487–2495.
[121] H. Bin *et al.*, "High-quality face image sr using conditional generative adversarial networks," *ArXiv*, Jul. 2017.
[122] D. Pathak *et al.*, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA*, 27-30 Jun. 2016, pp. 2536–2544.
[123] RA. Yeh *et al.*, "Semantic image inpainting with deep generative models," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR), Honolulu, HI, USA*, 21-26 Jul. 2017, pp. 5485–5493.
[124] C. Vondrick *et al.*, "Generating videos with scene dynamics," *Adv. in Neural Inf. Process. Syst.*, vol. 29, 2016.
[125] J. Wu *et al.*, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," *Adv. in Neural Inf. Process. Syst.*, vol. 29, 2016.
[126] C. Wang *et al.*, "Evolutionary generative adversarial networks," *IEEE Trans. on Evol. Comput.*, vol. 23, no. 6, pp. 921–934, Jan. 2019.
[127] I. Vaccari *et al.*, "A generative adversarial network (GAN) technique for internet of medical things data," *Sensors*, vol. 21, no. 11, p. 3726, May 2021.



B. Yousseu Tchaleu received her Bachelor's and Honor's degrees in Computer Science from the University of Johannesburg, South Africa, in 2015 and 2017, respectively. She received the M.Eng. degree in Electrical and Electronic Engineering with specialization in Artificial Intelligent from the same university in 2019 with Cum Lauder. Her master discussed effective algorithms to predict customer churn in financial services. At present, she is pursuing a PhD degree at the same university. From September 2021 to August 2022, she was a Data Scientist at iX-Engineer, Pretoria, South Africa, and from 2016 to 2018, she was a Software Engineer at MyBusinessSpace, Johannesburg, South Africa.

Alain R. Ndjiongue [S'14, M'18, SM'20] received his M.Eng. and D. Eng. Degrees in electrical and electronic engineering from the University of Johannesburg, South Africa, in 2013 and 2017, respectively. He was a Senior Lecturer at the same university from 2017 to 2020. He also served as a Senior Researcher at Memorial University of Newfoundland, Canada. From 2022 to 2024, he served as an Assistant Professor at the same institution. Since September 2024, he has been an Associate Professor at the University of the Witwatersrand, Johannesburg, South Africa. His research interests are in digital communications, including wireless, free-space optical, and visible light communications. He also researches optical reconfigurable intelligent surfaces, physical layer security in optical wireless communication systems, and artificial intelligence-aided systems. He is an Associate Editor of IEEE Communications Letters and a Senior Reviewer of the IEEE Open Journal of the Communications Society. He is also an active reviewer for several prestigious IEEE ComSoc journals and conferences. He was a 2017, 2018, and 2022 exemplary reviewer, and 2024 exemplary editor of IEEE Communication Letters, Optical Society of America. He is a 2019 top 1 percent peer reviewer in Essential Science Indicators Research.

Collins A. Leke received his M.Eng. and D. Eng. Degrees in electrical and electronic engineering with a speciality in Artificial Intelligence from the University of Johannesburg, South Africa, in 2014 and 2018, respectively. He was a Lecturer at the same university from 2019 to 2022. From 2022 until now, he is serving as a Senior Lecturer at the same institution. His research interests are in artificial intelligence, including machine learning, deep learning, and computational intelligence. He also researches generative adversarial networks , missing data in a variety of fields, artificial intelligence-aided systems, and swarm intelligence. He is an active reviewer for several journals and conferences.

CONTENTS PAGE