Generative adversarial networks: Core principles, cutting-edge models, broad applications, and contemporary challenges

Xiangzhong Ye

College of Computer Science and Technology, Zhejiang University, Hangzhou, 310058, China

3200104413@zju.edu.cn

Abstract. Generative adversarial networks stand out as one of the most notable innovations in the field of artificial intelligence. Often lauded for their capacity to emulate specific data distributions, their primary function is to discern the underlying characteristics of these distributions and subsequently generate data that mirrors them. In the realm of computer vision, GANs have showcased remarkable prowess by producing high-quality, realistic content. This capability has not only bolstered their reputation but also expanded their applicability across a multitude of tasks. However, the ascendancy of GANs isn't without its set of challenges. Training them can often be a delicate balancing act, as they require careful tuning to ensure stability. Issues like mode collapse, where the generator produces limited varieties of outputs, or training instabilities are not uncommon. Nonetheless, the inherent scalability and versatility of GANs continue to captivate researchers, making them a hotspot for innovation. As we delve deeper into the AI epoch, the potential of GANs remains vast, presenting both unprecedented opportunities and challenges.

Keywords: Generative Adversarial Networks, Foundational Theories, Advanced Models.

1. Introduction

Generative adversarial networks, introduced by Ian Goodfellow in 2014, have become a focal point in the AI research landscape. Celebrated for their unparalleled generative abilities and their versatility across various fields, GANs are a hotbed of contemporary research [1].

Supervised learning remains the most prominent and influential paradigm in machine learning. In this approach, given a sizable dataset with paired input-output examples, the algorithm discerns the relationship between inputs and their corresponding outputs, establishing a robust mapping between the two. This methodology excels particularly in classification tasks, where intricate data like images are mapped to distinct categories or labels, often represented as integers. Nevertheless, the Achilles' heel of supervised learning is its reliance on vast labeled datasets. Gathering such datasets, especially for niche tasks, can be both labor-intensive and costly due to the need for human annotation. Employing supervised learning on inadequate datasets can result in suboptimal performance and the dreaded overfitting. Contrastingly, unsupervised learning is tailored for unlabeled datasets. At its core, unsupervised learning seeks to unearth underlying patterns, structures, or distributions inherent in the data. Various algorithms under this umbrella aim to achieve distinct objectives based on the nature of the data and the task at hand, be it clustering, dimensionality reduction, anomaly detection, or denoising.

Generative modelling, as a broadly utilized unsupervised learning approach. Generative modelling assumes that the given training example $\boldsymbol{x}$ are drawn from an unknown but deterministic distribution $p_{data}(\boldsymbol{x})$. The goal of generative modelling is to learn a new distribution $p_{model}(\boldsymbol{x})$ that as close to $p_{data}(\boldsymbol{x})$ as possible. Therefore, generative modelling algorithms can generate examples from an approximated distribution of real dataset, which means they can augment the training dataset to enhance the performance of supervised learning and mitigate overfitting issues.

In today's digital age, GANs, a category of generative modeling algorithms, are making substantial strides across diverse sectors, from image processing and computer vision to texture synthesis and even Natural Language Processing (NLP). At its essence, the concept of GANs boils down to two intertwined networks engaged in a competitive dance. However, the sheer volume of research papers released annually—each detailing improvements, specialized variants, and novel applications of GANs—attests to the model's complexity and adaptability. Navigating the expanse of GANs-related research has become an intricate endeavor. This paper aims to simplify this task. In Section 2, we'll delve into the bedrock principles that underpin GANs. Section 3 will spotlight prominent GANs variations, categorized by their unique enhancement objectives. Section 4 will traverse the vast landscape of GAN applications. We'll dissect the strengths, weaknesses, and challenges posed by GANs in Section 5. Finally, Section 6 will wrap up our discussion with conclusions and contemplations on the future trajectory of GANs.

2. Foundational Theories of GANs

In this section, we first introduce related works and core concepts behind the original GANs, which followed by an introduction to mathematical frameworks of GANs.

2.1. Generative Modeling

There are primarily two types of generative models: explicit density models and implicit density models [1]. GANs belongs to the latter category, with numerous advantages over other similar algorithms in model performance and computational cost.

2.1.1. Explicit Density Models. Explicit density models assume that the data distribution can be approximated using parameterized probability distributions. These models explicitly define and parameterize a probability distribution $p_{data}(x, \theta)$, such as a Gaussian distribution, Bernoulli distribution, or a mixture of distributions. Then these models learn from true example input to determine its parameter θ , i.e., to capture the underlying structure and characteristics of the data [2]. Explicit density models not only can generate examples from approximated distribution, but also can estimate explicit distribution of given datasets.

Some common examples of explicit density models include Gaussian Mixture Models (GMMs), Kernel Density Estimation (KDE), Maximum Likelihood Estimation (MLE) and Markov Chain Monte Carlo (MCMC) [3]. Due to the characteristics of Explicit Density Models, although they behave well in certain situations, these methods have some inevitable limitations. For instance, GMM struggles with non-multimodal Gaussian distributions, KDE and MCMC usually entail high computational costs, MLE is prone to overfitting with complicated example inputs, and they all face challenges in handling high-dimensional data and are sensitive to hyperparameter choices [4]. This implies that as a category of generative modeling algorithms, explicit density models are hard to be used in problems like image processing or computer vision.

2.1.2. Implicit Density Models. Implicit density models, in contrast to Explicit Density Models, do not attempt to directly model the probability density function of data. Instead, implicit density models try to learn an underlying and unobservable data distribution, usually by learning generative function or mechanism [5]. Thus, implicit density model algorithms pay more attentions to the way of example generating, typically a mapping from random noise input into data examples. Implicit density models cannot estimate explicit distribution of given datasets but are good at generating complicated data examples compared to explicit density models. As a kind of implicit density models, the main advantage of GANs is to avoid introducing Markov Chain-based sampling, which is thought to be inefficient and has limitations in practical applications.

2.2. Core Concepts of GANs

The core concept of GANs is that two neural networks engaged in adversarial learning. A generator network endeavours to produce examples that resemble real examples, while a discriminator network strives to distinguish between the generated examples and real examples. This adversarial learning process propels both the generator and the discriminator to continually improve their capability of generating and discriminating [6]. After training, the generator is expected to be capable of generate examples that are too similar with real examples to be distinguished.

In the view of implicit density models, by modelling high-dimensional distributions of data implicitly, the generator of GANs strives to capture the distribution of true data samples and generate new data samples according to this distribution. Meanwhile, the discriminator, which is usually a binary classifier, is trained to discriminate generated samples as accurate as possible during the learning process. As shown in Figure 1.

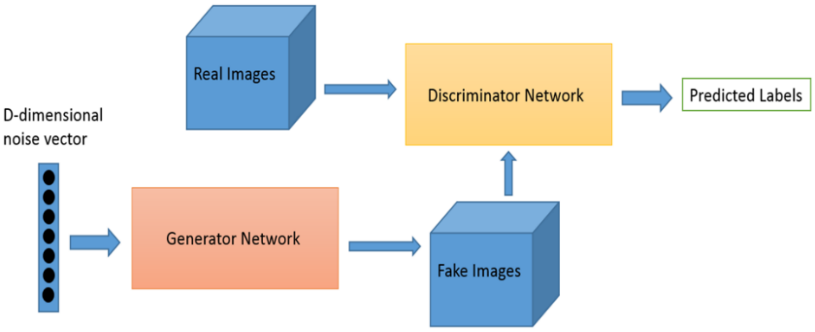

Figure 1. The general architecture of GAN [8].

As the general architecture of original GANs shown in Fig.1, the generator is a network that mapping random noise into generated examples (fake images). The discriminator is a network that discriminate fake images from real images. As for the generator network, the learning process is unsupervised and the feedback for parameters' adjustment comes from the discriminator. As for the discriminator network, the learning process is supervised because labels for real images and fake images are already known [7].

Generally speaking, parameters of two networks will be updated iteratively during the learning process to avoid instability and enhance optimization [8]. For instance, first parameters of the generator network will be fixed for the discriminator network to learn to update its parameters, and vice versa. Thus, GANs only update one of its networks for a single step.

2.3. Objective Functions

Mathematically, the optimization problem of GANs is considered to be a minimax problem, minimizing the generator's error while maximizing the discriminator's accuracy. In terms of game theory, the optimization objective of GANs is to achieve a Nash equilibrium between the generator and the discriminator.

The generator and the discriminator of GANs are both neural networks, which can be denoted as two differentiable functions G and D [9]. The input random noise is defined as z and the generated examples is denoted as $G(z)$. The outcome of the classifier network D over any example x is denoted as $D(x)$, which is either 0 or 1. $p_{data}(x)$ and $p_g(x)$ are distributions of real examples and approximated distributions of models. The objective function of original GANs is

$$\min_G \max_D V(D,G) = E_{x \sim p_{data}(x)}[log(D(x))] + E_{z \sim p_g(z)}\left[log\left(1 - D(G(z))\right)\right] \quad (1)$$

where the expectation term is typically approximated by sample minibatch of m samples at the same time and then update the discriminator and the generator with their stochastic gradient.

In the original paper, Ian Goodfellow has proved the objective function (1) is related with Jensen-Shannon divergence (JS divergence) between two probabilistic distributions $p_{data}(x)$ and $p_g(x)$, which indicates that the sufficient and necessary condition for the global optimum of a GANs is $p_{data}(x) = p_g(x)$ $[10]$. This mathematical proof shows that this form of objective functions makes GANs an implicit density model. Recently, researchers find assorted objective functions that can be used instead of JS divergence and improve the performance of GANs, which will be discussed in Section 3.

3. Variants of GANs

There are new GANs variants coming out every week with certain improvements on specified applications. It is really hard to keep track of them so this section will introduce some representative models of GANs. Generally speaking, there are mainly two ways researchers adopt to modify the original GANs. Some papers engaged in proposing different structures of GANs to solve certain issues or fulfill practical requirements. Others pay attention to the optimization approaches of GANs. They usually don’t change the core of the original GANs, but enhance it with carefully selected activation functions, neural networks and other optimization strategies.

3.1. Structural Diversities in GANs

The most common way to modify the original GANs is to change its structure. The original GANs have been extended by some researchers with additional inputs of the generator and outputs of the discriminator. Besides, some papers work on additional networks’ number and different network’s layouts of GANs. This kind of modifications is usually proposed in order to deal with practical requirements of application [2].

3.1.1. cGANs. The original GANs model is capable of generating examples from the distribution of input data. However, if input data contains different categories like different handwritten numbers, the original GANs can only generate images belong to handwritten numbers, where user can not choose which number to generate.

The conditional GANs (cGANs) extended the original GANs by adding conditional information y into the generator and the discriminator. The additional information y can be limited as exact handwritten numbers, categories of generated animals, certain limitations of generated images. It depends on the way dataset being labelled and how the training process utilize these labels. With conditional information y, the user can choose generated examples they need in some degree. The objective function is

$$\min_G \max_D V(D,G) = E_{x \sim p_{data}(x)}[log(D(x|y))] + E_{z \sim p_g(z)}\left[log\left(1 - D(G(z|y))\right)\right] \quad (2)$$

3.1.2. InfoGANs. The original GANs use a simple continuous noise input vector z and impose no restrictions on how the generator would use this noise. The generator may use this noise vector in a highly entangled way, implying that individual dimensions of z not correspond to semantic features of the data. However, it will be much more practical if semantic features can be controlled with input. For instance, in handwritten numbers case, it would be ideal that there is a single variable to control the generated number (0-9) and additional variables to control the stroke and angle of the number. Although the cGANs could behave well in this case, it requires additional supervision to specify labels of different examples, which is costly for larger dataset.

The Information Maximizing GANs (InfoGANs) are introduced to deal with the problem. Rather than using a single unstructured noise vector, it decomposes the input noise vector into two parts: z, the incompressible noise; c, the latent code [4]. While latent code is expected to be related with semantic features of real data distribution, the architecture of the original GANs promises that models can't treat two noise vectors in different ways. To ensure that the generated examples are saliently related with the latent code, InfoGANs utilize the concept of mutual information, which measures the correlation and the dependence between two random variables. The InfoGANs suggest that $I(c; G(z,c))$ should be high, which means that the latent code c has high correlation with the generated example $G(z,c)$. Therefore, the objective function of the InfoGANs is

$$\min_G \max_D V_I(D,G) = V(D,G) - \lambda I\big(c; G(z,c)\big) \quad (3)$$

where $V(D,G)$ is the objective function of the original GANs and λ is a tuneable parameter. In detail, InfoGANs usually not directly cope with the mutual information between the latent code and the generated example, but use mathematical technique of lower bounding to solve the formulation.

Although InfoGANs don't adopt specifical approaches to ensure the latent code c is disentangled, the experiment of the original paper shows that different variable of the latent code c are capable to represent different semantic features of the input data. As a unsupervised model, InfoGANs only need to determine how many semantic features it need to learn from the certain dataset before training.

3.1.3. CycleGANs. Domain adaption is a popular research direction in machine learning, which generally means that transform examples from a dataset to another related but different dataset with another distribution. For instance, transform realistic photographs into artworks with different styles is a typical domain adaption problem. Prior to Cycle-consistent GANs (CycleGANs), models that are used to solve image transforming problem require paired data (examples input and outputs), which is usually available in real datasets. CycleGANs can learn the mapping from original data distributions to target data distributions, which implies that it can be used on unpaired dataset.

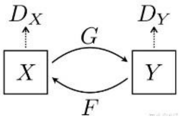

Figure 2. Structure diagram (Photo/Picture credit: Original).

The core concept of CycleGANs is to use a pair of GANs. There are two generators G and F, and two discriminators D_X and D_Y. As shown in figure 2, for the input image $x \in X$, the generator G transform it into $G(x) = y \in Y$ and vice versa. Tasks of two discriminators is to discriminate transformed images from real images as accurate as possible like that in the original GANs. With the architecture of CycleGANs, two generators can learn to transform images from one style into another, but in domain adaption problems, the content of two images are expected to be as similar as possible [6]. For example, if the realistic photograph shows a house, the transformed painting should show a house either. The CycleGANs suggest that there should be a generator F such that $F\big(G(x)\big) = x$ and introduce Cycle Consistency Loss to measure the difference between $F(G(x))$ and x. Although this is only the necessary condition of the correlation of contents, CycleGANs behave well in experiment.

3.2. Optimization strategies of GANs

The performance and stability of the original GANs used to be a problem. This kind of modification usually enhance the performance and stability of the original GANs to adopt a lager range of applications or simply accelerate convergence. Due to the fact that it is hard to theoretically evaluate the influence of modification like replacing a neural network with another one, these papers usually provide experimental evidences and reasonable hypothesis. Typically, these optimizations of GANs does not conflict with structural variations of GANs. For example, InfoGANs adopt Deep Convolutional GANs (DCGANs) to cope with image inputs and instability problems.

3.2.1. DCGANs. The original GANs training process is usually not as stable as anticipated, and the generator frequently ends up producing meaningless outputs or even crashing. Moreover, neural networks of the original GANs are fully-connected networks, which also means it's hard for the original GANs to deal with high-resolution images.

DCGANs replace two fully-connected networks of the original GANs with two deliberately designed CNNs (Convolutional Neural Networks). There are some other optimization strategies like adopting ReLU activations and using strided convolutions instead of pooling layers in DCGANs. All of these optimization strategies enable DCGANs to generate clearer image with sufficient stability.

3.2.2. f-GANs. The f-divergence is a universal framework to measure the difference between two certain distributions P and Q. For any function f such that $f(1) = 0$ and f is a convex function, it holds that

$$D_f(P||Q) = \int_x q(x) f\left(\frac{p(x)}{q(x)}\right) dx \tag{4}$$

is a divergence between two distributions P and Q. The goal of the GANs' optimization is to find $G^* = argmin_G\ D_f(P_{data}||P_G)$. Therefore, in f-divergence framework, different divergence functions D_f can be adopted by choosing convex function f for the optimization of the GANs in different situations [2]. For example, the function of $f(u) = ulog(u)$ induces $D_f$to be the Kullback-Leibler divergence (KL divergence). A deliberately selection of divergence can effectively deal with training problems of the GANs like mode collapse.

3.2.3. WGANs. Martin Arjovsky indicates that there are bugs in the design of the original GANs' objective function. For the theoretically optimal discriminator, the goal of training becomes minimize JS divergence between p_{data} and p_g. It has been proved that this kind of objective function will generally cause gradient vanishing problem. Therefore, if the discriminator's capability is trained so well, it is hard for the generator to obtain enough large gradient to continuously optimize itself.

Wasserstein GANs (WGANs) introduce a more proper mean to measure the distance between p_{data} and p_g, the Earth-Mover distance or simply the Wasserstein distance, which is defined as:

$$W(P_r, P_g) = \inf_{\gamma \sim \Pi(P_r, P_g)} E_{(x,y)\sim\gamma}[||x - y||] \tag{5}$$

It is hard to directly calculate Wasserstein distance, so the original paper turned to solving another formulation by utilizing Kantorovich-Rubinstein duality. The objective function of WGANs is:

$$\max_{w \in W} E_{x\sim P_r}[f_w(x)] - E_{p(z)}[f_w(g_\theta(z))] \tag{6}$$

where $\{f_w\}_w \in W$ is the goal of optimization, which is very similar to the discriminator of the original GANs. It has been proved that the more accuracy the new discriminator f_w has, the more improvement the generator will obtain.

4. Applications

The GANs and its variants are a kind of generative models, which can learn distributions from real dataset and generate examples from this distribution without explicitly expression. As expected, the GANs can generate more examples to help supervise learning. Additionally, these features allow GANs to become especially successful models in computer vision, including image and video processing. Recently, there are many developing applications of Human-Computer Interaction, Drug Discovery, Molecule Development. Section 3 has discussed many representative variant models of the GANs, but there are more application-targeted models derived from them.

4.1. Image-to-image Translation

As discussed earlier, CycleGANs is an important application model in image-to-image translation, whose mainly contribution is the removal of the limitation of paired data. The StarGAN is an extended model after CycleGANs, which allows transformation between many domains with a single generator. For example, StarGAN can transform a photograph of winter scene to both spring and summer scenes with different inputs.

4.2. Super-Resolution

Super-resolution is a category of method to upscaling images or videos with higher resolution. The traditional methods of super-resolution like Mean Square Error (MSE) will cause the problem of lacking high-frequency information. Super-Resolution GAN (SRGAN) apply the GANs framework in super-resolution tasks, which uses perceptual loss and adversarial loss as its objective function. Wang proposed Enhanced Super-Resolution GAN (ESRGAN) by improving the definition of perceptual loss and adversarial loss of SRGAN. They also change the discriminator into the Relativistic average Discriminator (RaD) based on the concept of Relativistic GAN, which helps improve the discriminator by dealing with its prior information of the equal number of real and fake examples [9]. Bulat indicates that most methods ignore how real word low-resolution images are produced. They propose to train another GANs model to learn how to transform high-resolution images to low-resolution images in a realistic way and then train SRGAN based on that realistic dataset produced by that high-to-low GAN.

4.3. Video Prediction and Generation

Video prediction and generation is a main problem of computer vision, which require algorithm to understand motions and objects. Mathieu firstly introduced GANs to predict frames after an input sequence. They proposed a combined strategies including GANs by training its discriminator to understand the information of time sequence. Based on Mathieu’s work, Ruben proposed a way to decompose motion and content, and then utilize GAN in training process.

Tulyakov proposed the Motion and Content decomposed GAN (MoCoGAN) for video generation. This model learns a mapping from a sequence of noise vector into a sequence of frames of a meaningful video. In detail, they decompose noise vector into a motion part and a content part, which is kept fixed for a certain video, by assuming there is no changes of contents.

5. Challenges and Future Work

The GANs as a generative model have been proven to have many advantages like high performance and costless computation. Recently, many papers emerging every year associated with GANs. For example, due to the contribution of WGANs, it is no longer necessary to carefully balance the training of the generator and the discriminator, avoiding overly small gradients that may lead to vanishing. However, there are still some fundamental issues that have not been completely solved.

The training process of the GANs is generally considered to be challenging. Mathematically, the loss function often fails to reliably converge to saddle points, resulting in poor model stability. On one hand, the model struggles to find the global optimum; on the other hand, it often exhibits parameter oscillations and fails to converge.

In the training process of GANs, mode collapse refers to the generator producing only a specific type of examples, such as generating only the number 1 in a handwritten number dataset, limiting the diversity of the generative model. Despite efforts in some papers to address these issues, mode collapse remains inevitably for complex high-dimensional data distributions. This discourage GANs to be used in those multi-modal problems due to its low diversity.

Recently, diffusion models have gained popularity as another type of generative model. They are capable of generating high-quality images, and their training process is highly stable, making them a potential competitor to GANs. It has been demonstrated that adjusting frameworks or applying various empirical optimization strategies is difficult to truly address the issues of training instability and mode collapse. To tackle these two challenging problems in the GANs field, as was done with the introduction of WGANs, future efforts may focus on establishing new mathematical paradigms to address the issues from a theoretical rather than empirical perspective.

6. Conclusion

Currently, the spotlight in GAN research is cast upon structural enhancements and refining optimization techniques. These modifications often serve dual purposes: they target specific application challenges while enhancing the model’s general capabilities. Structural tweaks, for instance, can be tailored to meet precise needs, such as rendering images at greater resolutions or producing artwork that captures diverse stylistic nuances. On the other hand, investigations into optimization strategies for GANs are paramount to enhance their overall performance metrics. These strategies primarily aim to bolster stability during the training phase and expedite the learning curve, ensuring that GANs deliver consistent and superior results. These nuanced improvements underscore the ever-evolving nature of GANs, demonstrating that while they are already formidable tools, there remains a wealth of untapped potential. Given GANs' foundational principles combined with their vast adaptability, the horizon seems boundless. The excitement in the research community is palpable, as every innovation marks a step closer to unlocking even more groundbreaking applications. As the journey of GANs continues, one can only anticipate a proliferation of remarkable advancements and transformative real-world implementations.

References

[1] Cai, Z., Xiong, Z., Xu, H., Wang, P., Li, W., & Pan, Y. (2021). Generative adversarial networks: A survey toward private and secure applications. ACM Computing Surveys (CSUR), 54(6), 1-38.

[2] Wu, A. N., Stouffs, R., & Biljecki, F. (2022). Generative Adversarial Networks in the built environment: A comprehensive review of the application of GANs across data types and scales. Building and Environment, 109477.

[3] Geowani, G., & Conri, M. (2023). Generative Adversarial Networks for Enhanced Learning Experiences.

[4] Gonog, L., & Zhou, Y. (2019, June). A review: generative adversarial networks. In 2019 14th IEEE conference on industrial electronics and applications (ICIEA) (pp. 505-510). IEEE.

[5] Chakraborty, T., Naik, S. M., Panja, M., & Manvitha, B. (2023). Ten Years of Generative Adversarial Nets (GANs): A survey of the state-of-the-art. arXiv preprint arXiv:2308.16316.

[6] Alqahtani, H., Kavakli-Thorne, M., & Kumar, G. (2021). Applications of generative adversarial networks (gans): An updated review. Archives of Computational Methods in Engineering, 28, 525-552.

[7] Lin, H., Liu, Y., Li, S., & Qu, X. (2023). How generative adversarial networks promote the development of intelligent transportation systems: A survey. IEEE/CAA Journal of Automatica Sinica.

[8] Alqahtani, H., Kavakli-Thorne, M., & Kumar, G. (2021). Applications of generative adversarial networks (gans): An updated review. Archives of Computational Methods in Engineering, 28, 525-552.

[9] Wang, Z., She, Q., & Ward, T. E. (2021). Generative adversarial networks in computer vision: A survey and taxonomy. ACM Computing Surveys (CSUR), 54(2), 1-38.

[10] Dou, B., Zhu, Z., Merkurjev, E., Ke, L., Chen, L., Jiang, J., ... & Wei, G. W. (2023). Machine learning methods for small data challenges in molecular science. Chemical Reviews, 123(13), 8736-8780.