ISSN:2790-1688 Volume-15-(2025)

Research on the Application of Generative Adversarial Networks in Artificial Intelligence Painting

Yue Xiao

Changzhou Senior High School of Jiangsu, Changzhou, Jiangsu province, China

468629061@qq.com

Abstract. GAN (Generative Adversarial Network) is widely used in image generation, renowned for its ability to produce high-fidelity details and sharp edges through adversarial training. Unlike Variational Autoencoders (VAEs), which often generate blurrier outputs, GANs excel in visual realism by leveraging a dual-network architecture—a generator and a discriminator—engaged in a competitive learning process. Furthermore, GANs synthesize images in a single forward pass, making them significantly faster than iterative approaches like Diffusion Models, which rely on multi-step denoising. This efficiency enables real-time applications, a critical advantage in fields such as AI-assisted art creation. This essay begins by outlining the foundational concepts of GANs, including their adversarial training mechanism. Next, it explores their methodology, emphasizing key architectures and training techniques that enhance stability and output quality. A comparative analysis with VAEs and Diffusion Models follows, highlighting GANs' superior perceptual quality while acknowledging challenges such as mode collapse and training instability. Finally, the discussion shifts to GANs' transformative role in AI painting, where they facilitate style transfer, photorealistic artwork generation, and interactive digital art tools. By examining these aspects, this essay underscores GANs' unique contributions to generative AI while addressing their limitations and future potential.

Keywords: Generative adversarial network, generation model, loss function, AI painting.

1. Introduction

Generative Adversarial Networks (GANs), first introduced by Ian Goodfellow et al. in 2014[1], represent a groundbreaking advancement in generative models within artificial intelligence. As a framework comprising two competing neural networks—a generator and a discriminator—GANs employ adversarial training to progressively enhance the generator's ability to produce highly realistic data samples.

What makes GANs truly revolutionary isn't just their technical specifications, it's how they've reshaped our entire approach to generative modeling. Unlike VAEs that play it safe with blurry approximations, or diffusion models that brute-force their way through hundreds of iterations, GANs dance on the edge of instability to achieve that magical combination of speed and quality. In the following part of the paper, we will discuss the methodology of GANs, their training mechanism, advantages and disadvantages, and their application in AI painting.

2. Background and Motivation

GAN was proposed to address the limitations of traditional generative models (e.g., VAEs, autoregressive models), such as blurry samples, low computational efficiency, and difficulty in modeling complex distributions. Through its unique adversarial training mechanism— pitting generator against discriminator in an AI "arms race"—GANs directly learn data distributions with human-like intuition, generating strikingly realistic and diverse samples that often fool even human observers, while overcoming traditional drawbacks of likelihood estimation and sampling inefficiency.

The core concept of GANs draws inspiration from the wisdom of human creative competition. Picture an ongoing duel between a young painter (the generator) and an art connoisseur (the discriminator)[2]: the painter continuously hones their craft, striving to create works so perfect they

could pass as masterpieces, while the connoisseur sharpens their eye to detect even the slightest flaws. Through this dynamic adversarial process, where each challenge makes the other stronger, the generator eventually learns to produce strikingly realistic creations. This adversarial philosophy mirrors how human skills evolve through competition, embodying the profound truth that "progress is born of opposition."

3. Methodology

3.1 GAN Architecture

The core and original function of GANs is **Adversarial Dynamics Formalization:**

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

As the function suggests, two models are trained in GAN: a generator and a discriminator. The function explicitly establishes the competitive relationship: The discriminator (D) maximizes the function to correctly classify real vs. fake samples, while the generator (G) minimizes the function to produce samples that fool D. At its heart, this adversarial battle continues until G's outputs become indistinguishable from real data—effectively mastering the art of perfect imitation.

The generator has the function of generating realistic data samples in an attempt to trick the discriminator into thinking they are real data. It receives a random noise vector as input and gradually converts it into an output that resembles real data through a series of neural network layers. And here is for the discriminator, the discriminator can distinguish between real data and synthetic data generated by the generator and decide whether the generated data is similar to the real data or not. It is achieved by receiving a data sample as input, processing it through a series of neural network layers, and outputting a probability value indicating the likelihood that the data is real.

GAN's basic architecture diagram is as follows:

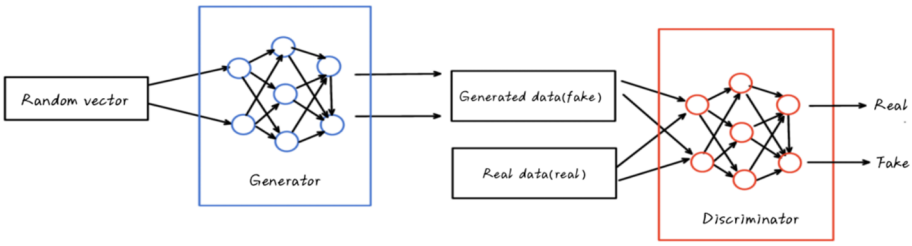

Figure 1: Basic architecture diagram of a Generative Adversarial Network [3].

According to the diagram, in conclusion, this is a game between a generator and a discriminator. The generator generates fake data, and then inputs both the generated fake data and real data into the discriminator, which needs to determine which ones are true and which ones are false. This process will not end until the generator's outputs become indistinguishable from real data.

3.2 Training Mechanism

GAN training is a dynamic game process: The generator (G) receives random noise to generate false samples, attempting to deceive the discriminator (D); The discriminator learns to distinguish between true and false by comparing real data and generated data. The two are alternately optimized-D improves discrimination by maximizing the accuracy of judging generated data, while G improves the level of falsification by minimizing D's ability to identify its generated samples. The training continues until both sides reach the Nash equilibrium, where the G-generated samples are almost indistinguishable from the real data distribution(I will explain it in the next two paragraphs). The core is to achieve data distribution matching through adversarial learning, without explicit modeling of probability density.

Mentioned in the previous paragraph, the Nash equilibrium plays the role as follows in GANs. At the Nash equilibrium, the generator's distribution p_g perfectly matches the real data distribution p_{data}, and the discriminator outputs $D(x)=0.5$ everywhere (maximal uncertainty). Neither can improve unilaterally since the system reaches a stable adversarial balance where optimal strategies mutually neutralize each other's advantage. Here $D(x)=0.5$ means no discriminative power.

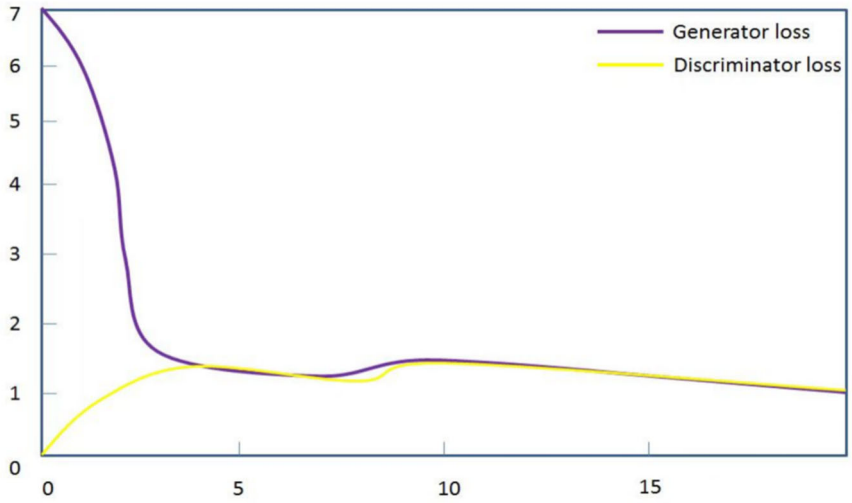

Figure 2: the generator loss and the discriminator loss [4].

This figure illustrates the adversarial dynamics en route to the Nash equilibrium in GANs. The generator's descending loss shows improving forgery skills, while the discriminator's peak-then-drop loss reflects their arms race. When both losses converge to zero, it mirrors the Nash equilibrium, where neither can outperform the other ($D(x)=0.5, p_g = p_{data}$), completing the adversarial learning objective. (The loss function in the image will be introduced in the next part.)

3.3 Loss Function

Loss function is an important concept in machine learning and deep learning. It serves as the fundamental driver of GAN training, acting as both referee and coach in the adversarial competition. For the generator, the loss function quantifies how convincingly its synthetic samples mimic real data, providing gradients to improve its "forgery skills." For the discriminator, the loss measures its ability to detect fakes, pushing it to become a better "art authenticator." This delicate balance—where each network's loss depends on the other one's performance—creates the dynamic tension that propels both toward excellence. Well-designed loss functions prevent training collapse, maintain stable gradients, and ensure diverse output generation. They essentially encode the rules of the adversarial game, determining whether the system achieves high-quality results or fails catastrophically.

Next, I will show a diagram to explain the loss function's basic architecture.

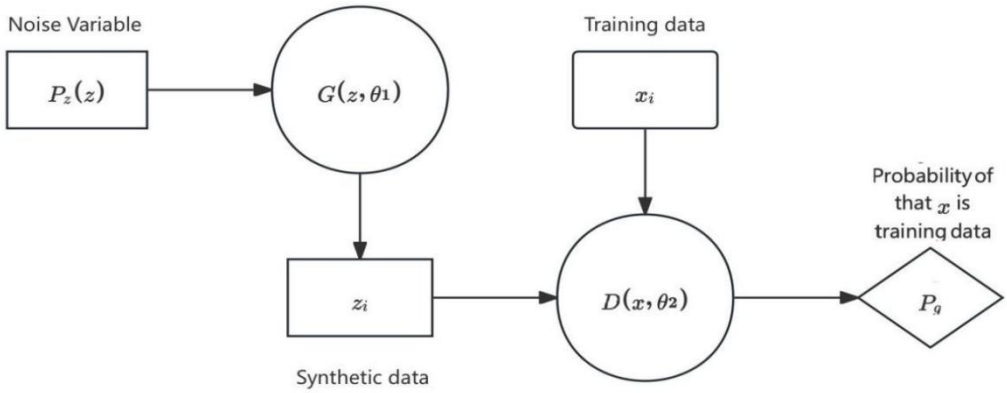

Figure 3: the loss function's basic architecture diagram [5].

This diagram illustrates the core workflow of a GAN: The generator (G) transforms noise variables $z \sim P_z(z)$ into synthetic data, while the discriminator (D) receives both generated and real data x_i, outputting a probability P_ϑ to distinguish their origins. The parameters ϑ_1 and ϑ_2 represent the trainable weights of G and D, respectively, forming an adversarial training loop.

In GANs, there are two primary types of loss functions that govern the adversarial training process[6]:

The first type of loss function is the loss function of a generative network:

$$L_G=-E[D(G(z))]$$

Here, H represents cross-entropy, and z is the input random data. D(G(z)) is the probability of judging the generated data, where 1 represents absolute truth and 0 represents absolute falsehood. The function represents the distance between the judgment result and 1. Obviously, in order to achieve good results in generative networks, it is necessary to have the discriminator distinguish the generated data as true data, which means the smaller the distance between D(G(z)) and 1, the better it is.

The second type is the loss function of discriminative network:

$$L_D=E[D(G(z))]-E[D(x)]$$

To achieve good results in network recognition, it is necessary to recognize that in its eyes, real data is real data, and generated data is fake data; that is to say, the distance between real data and 1 is small, and the distance between generated data and 0 is small as well.

In conclusion, loss functions serve as the core driver of GAN training, where their design directly determines model convergence, generation quality, and training stability—making careful selection critical for balancing the adversarial dynamics between generator and discriminator.

3.4 Training Challenges

Here is a training example of GANs:

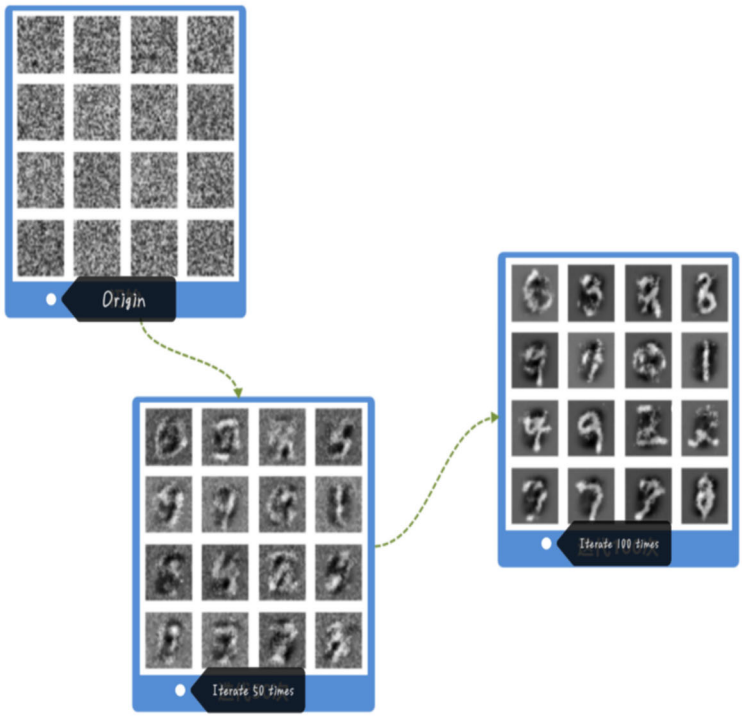

Figure 4. A training example of a Generative Adversarial Network (GAN) [7].

From the example, we can see that after iterations, the numbers that the machine generates seem clearer. However, the training of GAN requires high computing power and is unstable when training more complex problems, easily encountering problems such as gradient vanishing and pattern collapse, making the training difficult.

Firstly, with only 50-100 iterations shown, the model almost certainly hasn't completed the critical early phase where meaningful features emerge. GANs typically require 10K+ iterations to converge, and this would explain why outputs appear unstructured or noisy. Additionally, the process may also face the problem of mode collapse. The generator produces limited varieties of samples, ignoring full data diversity, by exploiting weaknesses in the discriminator. Common in early training phases without proper regularization techniques. Last, the GANs are notoriously sensitive to hyperparameter choices—slight changes in learning rates, batch sizes, or optimizer settings can dramatically impact training stability and output quality. Common pain points include unbalanced generator/discriminator learning rates and improper noise vector initialization.

3.5 Advantages and Disadvantages

Compared to traditional models, it has two different networks instead of a single network, and the training method used is adversarial training. Also, the gradient update information of G in GAN comes from the discriminator D, but not from the data sample.

The advantages of GAN are significant. Firstly, GAN is a generative model that uses backpropagation instead of complex Markov chains compared to other generative models such as Boltzmann machines and GSNs. Secondly, compared to all other models, GAN can generate clearer and more realistic samples. Thirdly, GAN is trained using an unsupervised learning approach and can be widely used in both unsupervised and semi supervised learning fields. Moreover, compared to variational autoencoders, GANs do not introduce any deterministic bias. Variational methods introduce deterministic bias because they optimize the lower bound of log likelihood rather than likelihood itself, which appears to result in VAEs generating more ambiguous instances than GANs. Finally, GAN is applied to some scenarios, such as image style transfer, super-resolution, image completion, and noise reduction, avoiding the difficulties of loss function design.

However, some disadvantages still exists. Despite their impressive capabilities, GANs suffer from several critical drawbacks. First, their training process is highly unstable due to the challenge of reaching the Nash equilibrium, making them less reliable than alternative models like VAEs [8] or PixelRNN [9]. Additionally, GANs struggle with discrete data such as text—since text is typically represented as one-hot vectors, minor variations in generator outputs often fail to produce meaningful gradient updates, causing training to stagnate. Additionally, the use of JS divergence as a loss function is problematic, as it performs poorly when comparing distributions with minimal overlap. Finally, early GANs faced issues like unstable training, vanishing gradients, and model collapse—where generated samples remained poor despite prolonged training. This occurs because G's updates rely on D's feedback: if D misjudges weak samples as realistic, G reinforces those flaws, leading to self-deceptive degradation and incomplete feature generation. These limitations restrict GANs' effectiveness in certain applications and remain active research challenges.

4. Comparison with Other Generative Models

4.1 Comparison with Variational Autoencoder (VAE)

The core idea of VAE [4] is to view the generative model as a combination of encoder and decoder. The encoder compresses the input data into low-dimensional random variables, while the decoder decodes these random variables into estimates of the original data.

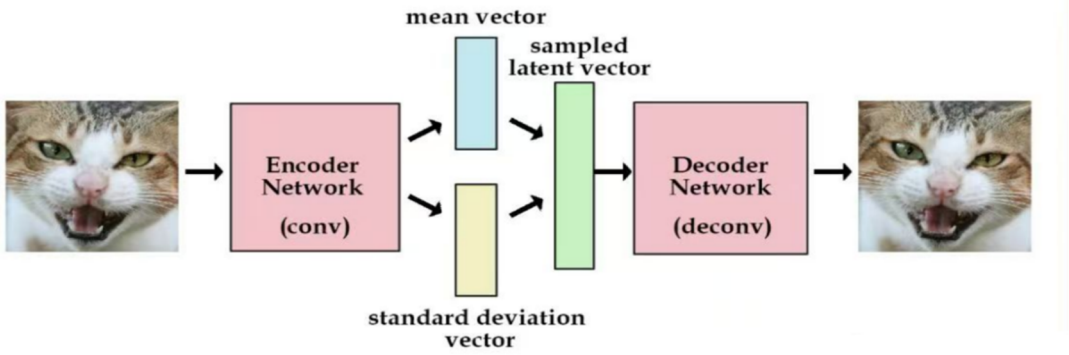

Figure 5: VAE's basic architecture diagram [10].

Therefore, in an ideal situation, the restored output image should be very similar to the original image. However, compared to GAN, the images generated by VAE are less vivid since the GNA's optimization goal is to make the picture realistic. But from another point of view, the diversity of images generated by VAE is much more sufficient than that of GAN, due to the stronger probability distribution of learning and interpretability.

4.2 The comparison with the Diffusion Model (DM)

DM is another generative model that generates data through a gradual noise addition and denoising process[11]. The theory first defines the Markov chain of diffusion steps to slowly add random noise to the data, and then learns the backwards diffusion process to construct the required data samples from the noise, just like the diagram shows below.

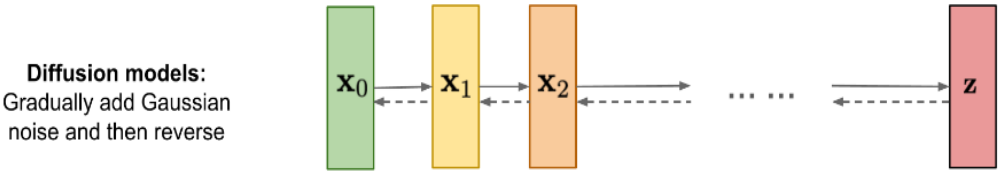

Figure 6. Basic architecture diagram of a Diffusion Model (DM) [12].

Diffusion Models are generative models that learn to synthesize data by first progressively corrupting training samples with Gaussian noise (forward process) and then training a neural network to reverse this noising process (reverse process). The model's latent space maintains the same dimensionality as the input data through a Markov chain structure, enabling it to effectively generate new samples by denoising random noise inputs. This approach combines the stability of iterative refinement with strong theoretical foundations from stochastic processes.

Compared to GANs, Diffusion Models offer several differences: training difficulty, simulation distribution continuity, and simulation controllability. Firstly, GANs and Diffusion Models exhibit fundamental differences in training stability. While GAN training is notoriously unstable, often suffering from mode collapse and oscillating losses, Diffusion Models demonstrate significantly better convergence with smooth, predictable loss curves that enable more reliable optimization.

Secondly, their approaches to distribution modeling differ substantially. Diffusion Models excel at capturing complex, nonlinear distributions across diverse datasets, which explains their dominance in large-scale generative systems like Stable Diffusion. However, this capability comes at the cost of temporal continuity in applications like video generation. In contrast, GANs achieve superior performance on homogeneous datasets (e.g., face generation) but struggle when handling multi-category image collections due to their limited distribution modeling capacity.

Finally, their controllability mechanisms present distinct advantages. GANs (exemplified by StyleGAN) offer direct manipulation through interpretable latent spaces (w-space), enabling precise edits like continuous interpolation and DragGAN's pixel-level control. Diffusion Models (e.g., Stable Diffusion[13]) implement control through text embeddings, providing powerful semantic manipulation via prompt engineering, though this approach offers less explicit control over the latent space compared to GANs' more structured frameworks.

5. The application of GAN in AI painting

AI painting has emerged as a groundbreaking technology in recent years, sparking both excitement and debate. While some fear it may threaten traditional painting professions, others recognize its potential to inspire artists with innovative tools, enabling them to create more imaginative works. Moreover, AI painting democratizes art by allowing anyone—regardless of skill—to generate stunning visuals simply by describing their vision in text, offering a deeply rewarding creative experience.

The AI painting process involves five key steps: First, data collection gathers diverse artworks across styles, artists, and themes to train the model on artistic variety. Next, preprocessing cleans and normalizes the data, including resizing images, standardizing pixel values, and applying noise reduction techniques. For instance, neural networks can transform low-sample (e.g., 4spp) noisy inputs into high-quality images. Then, model training iteratively optimizes the algorithm using curated data and loss functions until it can convincingly replicate the training dataset's features. Following this, art generation produces new works by feeding random noise or specific latent vectors into the trained model. Finally, post-processing refines the output, addressing issues like uneven exposure. Techniques such as histogram equalization enhance contrast by redistributing pixel values, making images more visually cohesive and realistic.

Through these steps, GANs empower AI painting to merge technical precision with artistic expression, reshaping how art is created and appreciated. While challenges remain, the technology's ability to generate impressive, personalized artwork heralds a transformative era for both artists and enthusiasts.

6. Development of GAN in AI painting

Since its birth in 2014, the development of GANs has been very rapid. Initially, the early GANs were difficult to generate high-quality images, but now GANs can already generate high-quality

images. In 2015, DCGAN was proposed, combining GAN with CNN. For example, these images are the ones that DCGAN generated. Later in 2016, CycleGAN and pix2pix emerged, which could achieve image-to-image translation. And by 2018, ProGAN and StyleGAN (which is based on ProGAN[14]) were developed and became widely spread and used on the Internet.

7. Conclusion

Generative Adversarial Networks (GANs), as an important branch of the deep learning field, have a very promising future. With the improvement of computing power and the continuous progress of algorithms, the application of GANs in fields such as image generation, style transfer, data augmentation, and image restoration will become more profound.

In the future, GANs are expected to make significant progress in the following areas: 1. Media Generation: Enhanced model architectures will enable GANs to produce ultra-realistic, high-resolution images and videos, pushing the boundaries of synthetic media. 2. Learning Paradigms: Their ability to leverage unlabeled data will revolutionize unsupervised and semi-supervised learning, improving data distribution modeling. 3. Cross-Modal Applications: GANs will bridge modalities (e.g., text-to-image), driving innovations in machine translation and generative content creation. 4. Healthcare Innovations: In medical imaging and bioinformatics, GANs will enhance data augmentation and diagnostic accuracy through synthetic data generation. 5. Security Solutions: Dual-use capabilities include privacy protection via anonymized data synthesis and robustness testing through adversarial example generation. 6. Creative Industries: The entertainment sector will adopt GANs for art generation, game asset design, and immersive VR experiences, expanding creative possibilities.

Despite the promising future of GAN development, there are also challenges, such as the stability and controllability of model training, and ethical and legal issues related to the generated content. Future research needs to achieve a balance and breakthrough in these areas to ensure the healthy development and widespread application of GAN technology.

References

[1] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial networks. *Advances in Neural Information Processing Systems*, *27*, 2672–2680.

[2] Wang, A., Liu, R., Liu, X., Chen, J., & Liao, Q. (2020). GANs as creative adversaries. Proceedings of the AAAI Conference on Artificial Intelligence, 34*(04), 6109-6116. https://doi.org/10.1609/aaai.v34i04.6064

[3] CSDN. (2024). Deep learning: GAN training examples. CSDN Blog. Retrieved from: https://so.csdn.net/

[4] CSDN. (2024). Generative Adversarial Network (GAN) architecture diagram. CSDN Blog. Retrieved from https://so.csdn.net/

[5] Goodfellow. (2014). Generative adversarial networks.

[6] Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein GAN. *Proceedings of the 34th International Conference on Machine Learning* (ICML 2017), *70*, 214-223.

[7] CSDN. (2022). Training examples of a Generative Adversarial Network (GAN). CSDN Blog. Retrieved from: https://so.csdn.net/

[8] Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[9] Van den Oord, A., Kalchbrenner, N., & Kavukcuoglu, K. (2016). Pixel recurrent neural networks. *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 48, 1747–1756.

[10] CSDN. (2024.01).GAN&VAE& . CSDN Blog. Retrieved from https://so.csdn.net/

[11] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 6840–6851.

[12] CSDN. Diffusion Model (DM) architecture diagram. CSDN Blog. Retrieved from: https://so.csdn.net/

[13] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10684–10695.

[14] Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation. *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, 30, 1646–1654.