A Primary Comparison of Diffusion Models and Generative Adversarial Networks for Image Synthesis

Zhuoyi Shen
Department of Intelligent Science, Xi'an Jiaotong-Liverpool University
China
zhuoyi.shen20@student.xjtlu.edu.cn

Maoyu Mao
Department of Intelligent Science, Xi'an Jiaotong-Liverpool University
China
maoyu.mao20@student.xjtlu.edu.cn

Pengfei Fan
Department of Intelligent Science, Xi'an Jiaotong-Liverpool University
China
School of Electronic Engineering and Computer Science, Queen Mary University of London, London
UK
pengfei.fan@xjtlu.edu.cn

Abstract

The aim of this paper is to investigate the application of different types of datasets on image generation models, specifically the MINIST dataset and the CIFAR-10 dataset, and experiments were conducted using Diffusion Models and Generative Adversarial Networks (GANs) models. The performance and training process are evaluated and analyzed by comparing the two generative models for image synthesis. Through these comparison experiments, we find that both models have impressive performance in image generation. Specifically, Diffusion Models have a more stable training process and perform better in the later stages of training, while GANs have a shorter training time but are relatively less stable due to their adversarial training approach, and have more prominent generation results in the early stages but are slightly weaker than Diffusion Models in the later stages. These findings help me better understand and compare the characteristics and applicability scenarios of different generative models and applicable scenarios.

CCS Concepts

• **Image Generation**; • **Enhancement**; • **Restoration**;

Keywords

Diffusion Models, Generative Adversarial Networks, Image Synthesis, Conditional Generative Models

ACM Reference Format:
Zhuoyi Shen, Maoyu Mao, and Pengfei Fan. 2024. A Primary Comparison of Diffusion Models and Generative Adversarial Networks for Image Synthesis. In *2024 The 7th International Conference on Machine Learning and Machine Intelligence (MLMI) (MLMI 2024), August 02–04, 2024, Osaka, Japan*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3696271.3696307

MLMI 2024, August 02–04, 2024, Osaka, Japan

ACM ISBN 979-8-4007-1783-3/24/08
https://doi.org/10.1145/3696271.3696307

1 Introduction

Image generative modeling is a significant research direction in artificial intelligence, aiming to generate high-quality, diverse images that mimic human-observed patterns. With advances in deep learning and generative modeling, image generation techniques have shown great potential and applications in various fields.

In recent years, generative models have made significant strides in image generation, evolving from pixel reconstruction methods to advanced techniques like Generative Adversarial Networks (GANs) and Diffusion Models. These developments have achieved remarkable success.

The background of image generation model development is multifaceted. The widespread use of digital images in social media, virtual reality, and medical imaging has created a growing demand for high-quality images. Additionally, advancements in image generation technology provide creative opportunities in art, film, special effects, and game development. Furthermore, these models have practical applications in medical image analysis, autonomous driving, and security monitoring.

Thus, research in image generation models is not only academically significant but also holds extensive application prospects in industry and social life. In low-level vision, generative models play a crucial role by generating new data through analysis of existing datasets' underlying distributions [1]. This report focuses on deep generative models using neural networks to capture complex patterns in probability distributions, enabling the generation of realistic and diverse data. Training these models usually requires large datasets; limited data can hinder the model's ability to learn data diversity and characteristics, affecting the quality of generated data. To mitigate this, methods like GANs and diffusion models have been developed. However, there is a lack of comprehensive research comparing these two models. Therefore, this study focuses on comparing and evaluating these models in the context of image generation, aiming to provide deeper insights into their respective strengths and weaknesses. In this study, the following contributions were made: 1) Literature Review: Reviewed GANs and Diffusion Models to provide a theoretical foundation; 2) Model Implementation: Implemented Denoising Diffusion Probabilistic Models (DDPM), Conditional Diffusion Models, and Auxiliary Classifier Generative Adversarial Networks(AC-GANs) frameworks; 3) Dataset Selection: Used MNIST for DDPM and CIFAR-10 for Conditional Diffusion Models and AC-GANs; 4) Image Analysis:

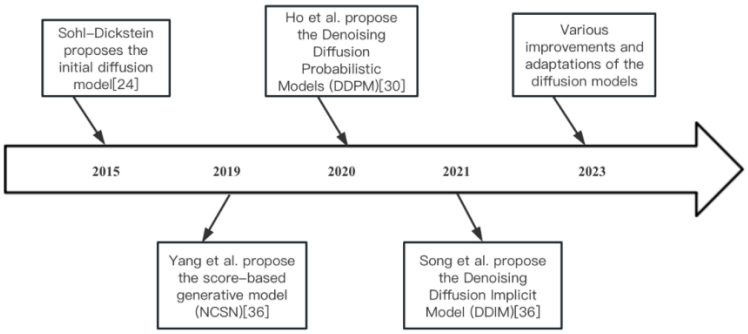


Figure 1: The representative works in diffusion models for different applications.

Analyzed and compared generated images for quality, diversity, and stability to assess model strengths and weaknesses.

2 model description

2.1 Generative adversarial networks

GANs consist of two competing networks: the generator and the discriminator. The generator aims to create samples that deceive the discriminator, which in turn tries to distinguish between real and generated samples. This adversarial process allows the generator to progressively produce more realistic images [1–4].

Since Ian Goodfellow introduced GANs in 2014, they have been applied in fields such as image and video generation, data enhancement, and style transformation [1]. In image generation, GANs have made synthesized images clearer and more natural. Notable examples include CycleGAN [5], Pix2Pix GAN [6], and Conditional GAN [7]. CycleGAN introduces cyclic consistency loss for unpaired transformations, while others like Ledig et al. [9] and Wu et al. [10] have shown the potential of GANs in generating high-resolution images with sharp edges and rich details. Wang et al. [11] used residual blocks and relative GANs to improve visual quality, while Wang et al. [12] synthesized high-resolution images from labeled graphs using conditional GANs. In image enhancement, Antoniou et al. [13] demonstrated that Data Augmented GANs (DAGANs) effectively augment standard classifiers. In style transformation, Gatys et al. [14] introduced StyleGAN, which combines artworks with arbitrary images to achieve style transfer.

However, GANs face challenges such as training instability and pattern collapse, leading to poor image quality, especially with high-resolution datasets [2, 15–20]. These issues arise due to various factors, including generator architecture, loss functions, and distance metrics [17]. During iterative training, instability can misclassify real samples, while pattern collapse limits the variety of generated samples. GANs also suffer from catastrophic forgetting, where the discriminator fails to convert real data points into local maxima, preventing GAN training from converging [18].

Despite being state-of-the-art, these drawbacks limit GANs' application to new areas, prompting research into likelihood-based models to achieve GAN-like sample quality [21–23, 29].

2.2 Diffusion models

Diffusion models, characterized by likelihood-based modeling and progressive random sampling, have emerged as promising alternatives to GANs for high-quality image generation [24–26, 29]. They offer desirable properties like distributional coverage, static training targets, and easy scalability, improving sample fidelity in unconditional generative tasks [31].

Diffusion probabilistic models transform complex generative processes into stable inverse processes using Markov chain models [27]. Typically, a diffusion model includes a forward process, which gradually adds noise to an image until it conforms to Gaussian noise, and an inverse process, which denoises and reconstructs the image by predicting noise or estimating the score [27–29].

As Figure 1 shows, the development of the diffusion model progresses through several stages.

In 2015, Sohl-Dickstein proposed diffusion models to address the challenges faced by generative models like VAEs, which require simultaneous optimization of the field distribution and variational posterior. Diffusion models simplify this by mapping data distribution to a standard Gaussian, making the generator's task easier by fitting each small step of the inverse process [24]. In 2019, Yang et al. introduced a score-based generative model (NCSN), achieving GAN-level sample quality without adversarial training and with a more flexible architecture [25, 36]. In 2020, Ho et al. presented Denoising Diffusion Probabilistic Models (DDPM), combining diffusion models with denoising scores to guide training and sampling, resulting in improved image generation and a more stable training process [30]. In 2021, Song et al. proposed the Denoising Diffusion Implicit Model (DDIM), which extends traditional Markov diffusion to a non-Markov process, accelerating sample generation with fewer steps [36]. Stochastic differential equations (SDEs) form a third subclass of diffusion models, offering efficient generative strategies and powerful theoretical results [37, 38]. Saharia et al. introduced the SR3 method in 2021, enhancing resolution through repetitive refinement for more realistic outputs [39].

Despite their ability to generate high-quality images, diffusion models require multiple inference steps, making image generation relatively slow and limiting their applicability to large-scale and real-time scenarios [2, 30].

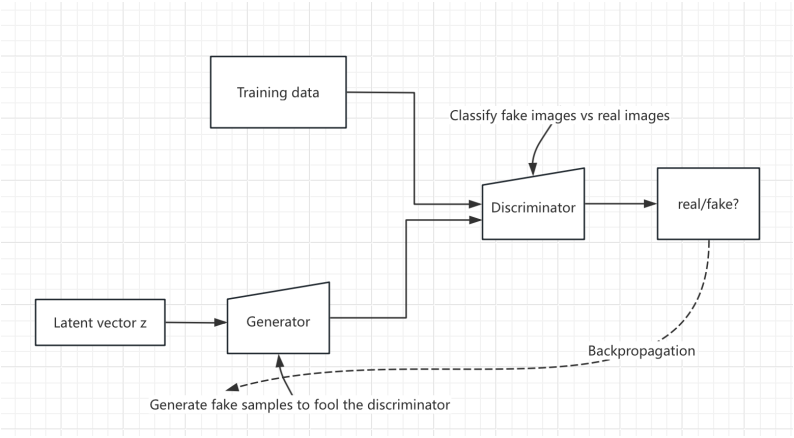


Figure 2: The principle of GANs.

3 model formulation

3.1 Generative adversarial networks

GANs consist of two components: the generator and the discriminator. The generator takes random samples from the latent space and produces outputs resembling real samples. The discriminator receives either real samples or generator outputs and attempts to distinguish between them. The generator's goal is to deceive the discriminator, while the discriminator aims to correctly classify samples. Through this adversarial process, both networks continuously adjust their parameters to make the discriminator unable to distinguish between real and generated samples.

The mathematics of GANs can be expressed as an adversarial min-max game. We define a value function V(G,D), where the discriminator D aims to maximize its ability to distinguish between real and fake samples, and the generator G aims to minimize this value, generating samples that are hard to distinguish. This process is formalized by the following objective function:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{11}$$

For ① part: Given G, find D that maximizes V

$$\max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{12}$$

At this point the x input to the discriminator is real data. A larger value of $E_{x\sim p_{data}(x)}[\log D(x)]$ indicates a higher probability that the discriminator considers the input x to be real data, i.e., the discriminator is more capable. Therefore the larger the input to this item the better it is for the discriminator. For the latter part of the above equation. At this point the input to the discriminator is the false image G(z). As $\log(1-D(G(z)))$, the smaller the value of D(G(z)), the larger the value of $E_{z\sim p_z(z)}[\log(1-D(G(z)))]$, it means the smaller the probability that the discriminator decides that the fake image is real data, i.e., the stronger the discriminator is.

For ② part: Given D, find G that minimizes V

$$\min_G V(D,G) = \underbrace{E_{x\sim p_{data}(x)}[\log D(x)]}_{1} + \underbrace{E_{z\sim p_z(z)}[\log(1-D(G(z)))]}_{2}$$

.

By fixing D, part 1 is only related to D, so it is constant and has no effect on minimizing V. For part 2, the goal now is that the generator works well, i.e., it conceals as much as possible from the discriminator, i.e., it is expected that D(G(z)) is as large as possible, at which point the input G(z) to the discriminator is a fake image. The larger D(G(z)) is, the higher the probability that the discriminator will decide that the fake image is the real data, which means that the generator generates a good image and can successfully fool the discriminator. And at this point, the smaller $E_{z\sim p_z(z)}[\log(1-D(G(z)))]$ becomes, so finding G that minimizes V will give the generator the best results.

3.2 Diffusion models

The model is mainly divided into forward process and reverse process, as shown in the Figure 3 below:

3.2.1 Forward Process. For a given model of an original photograph x_0, the model changes the picture from x_0 to x_1 by adding a Gaussian noise to it, and keeps repeating the above steps until the picture becomes x_n, approximately, x_n obeying a Gaussian distribution. The steps of adding noise can be represented by the following equation:

$$X_t = \sqrt{a_t}X_{t-1} + \sqrt{1-a_t}Z_1 \tag{1}$$

In the above formula, X_t denotes the image at the moment t, while thenX_{t-1} correspondingly denotes the image at the moment t−1, Z_1 denotes the Gaussian noise added at that moment, obeying the standard Gaussian distribution of N(0,1). And $\sqrt{a_t}$ and $\sqrt{1-a_t}$ denote the magnitude of the weights of these two quantities, the sum of squares is 1. From this formula, it can be seen that the image of the latter moment is determined by the image of the previous moment and the added noise.

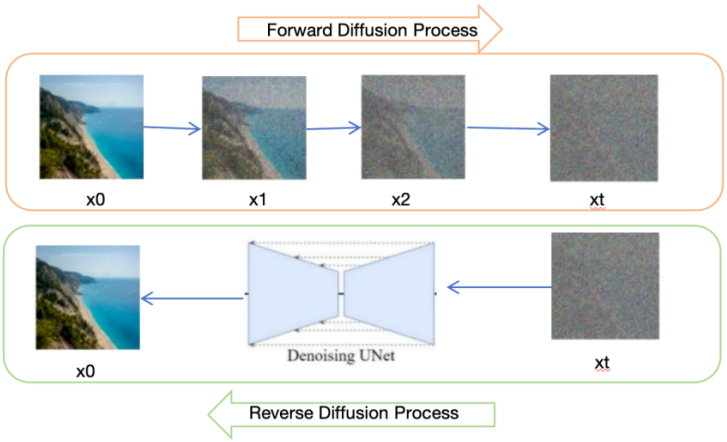


Figure 3: Graphical representation of forward and reverse processes. Modified from "Diffusion Model Clearly Explained! How does AI artwork work? Understanding the tech behind the rise of AI-generated art" by Steins, via Medium. https://medium.com/@steinsfu/diffusion-model-clearly-explained-cd331bd41166

And the magnitude of the weight(a_t) is related to another quantity β_t with the following formula.

$$a_t = 1 - \beta_t \tag{2}$$

β_t as a pre-given value, it is increasing with moments in the range [0.0001,0.02]. If β_t is getting bigger and bigger, a_t is getting smaller and smaller, then with $\sqrt{a_t}$ and $\sqrt{1-a_t}$ is getting smaller and bigger. According to 1), $\sqrt{1-a_t}$ as Z_1's weights are getting bigger and bigger, which means that more and more Gaussian noise is added as the steps increase.

In 1), the image of moment X_{t-1} needs to be obtained from the image of moment X_t. And the image of the moment of X_{t-1} needs to be derived from the image of the moment of X_{t-2}. With the subsequent derivation, the image of the moment of X_{t-3} leads to the image of the moment of X_{t-2}. Until the moment X_0 to derive the X_1 moment image. The formula is as follows:

$$\begin{aligned} X_t &= \sqrt{a_t}\left(\sqrt{a_{t-1}}X_{t-2} + \sqrt{1-a_{t-1}}Z_2\right) + \sqrt{1-a_t}Z_1 \\ &= \sqrt{a_t a_{t-1}}X_{t-2} + \sqrt{a_t\,(1-a_{t-1})}Z_2 + \sqrt{1-a_t}Z_1 \\ &= \sqrt{a_t a_{t-1}}X_{t-2} + \sqrt{1-a_t a_{t-1}}\widehat{Z_2} \end{aligned} \tag{3}$$

The last step of the above equation uses the relevant properties of the Gaussian distribution. According to Equation(4), the relationship between the X_{t-3} moment image and the X_{t-2} moment image can be derived as follows:

$$X_{t-2} = \sqrt{a_t a_{t-1} a_{t-2}}X_{t-3} + \sqrt{1-a_t a_{t-1} a_{t-2}}\widehat{Z_3} \tag{4}$$

According to 4), the relationship between the X_t moment image and the X_0 moment image can be derived as follows:

$$X_t = \sqrt{\overline{a_t}}X_0 + \sqrt{1-\overline{a_t}}\widehat{Z_t} \tag{5}$$

According to the above equation, the model can add noise to the original image X_0 by a set time step t to form a noisy image conforming to a Gaussian distribution.

3.2.2 Reverse process. The inverse process is the process of reducing the Gaussian noise to the expected picture. That is, the process of reducing the Gaussian noise image at moment X_t to the expected picture(X_0).

Here Bayesian formula is required and the formula is given below:

$$P(A\,|B) = \frac{P(B\,|A)\cdot P\,(A)}{P\,(B)} \tag{6}$$

And use the formula to find the X_{t-1}-moment image as follows:

$$q\,(X_{t-1}\,|X_t\,) = q\,(X_t\,|X_{t-1}\,)\,\frac{q\,(X_{t-1})}{q\,(X_t)} \tag{7}$$

The $q(X_{t-1}|X_t)$ in 7) can be found by a forward process that changes the equation to the following equation by increasing X_0:

$$q\,(X_{t-1}\,|X_t\,,\,X_0) = q\,(X_t\,|X_{t-1}\,,\,X_0)\,\frac{q\,(X_{t-1}\,|X_0\,)}{q\,(X_t\,|X_0\,)} \tag{8}$$

Through a series of calculations, the following formula can eventually be derived:

$$X_0 = \frac{1}{\sqrt{\overline{a_t}}}\left(X_t - \frac{\beta_t}{\sqrt{1-\overline{a_t}}}\widehat{Z}_t\right) \tag{9}$$

The mean and variance of the image at moment t − 1can be estimated and consequently the image at moment X_{t-1} can be estimated and consequently the image at moment X_0 can be obtained.

Based on the above formula, the model can generate an approximation of the original picture based on the Gaussian noise picture.

4 Implementation and Results

4.1 DDPM Implementation

During initialization, the model sets diffusion process parameters, including betas, alphas, and coefficients for a series of time steps. The "sample_forward" method generates an image with added noise, given an input time step. The "sample_backward" and "sample_backward_step" methods perform backward sampling to denoise the image. "Sample_backward" iteratively applies backward

Figure 4: Five random sets of noisy images from DDPM with time step (a) 0-50 and (b) 0-300 in MINIST

steps to produce the final image. Additionally, the "visualize_forward" method displays images of the noise addition process. Figures 4 show examples of the noise addition process at different step sizes (50 and 300 steps):

These images illustrate the progressive blurring during the noise addition process, giving a diffuse effect as the time step increases. The neural network architecture is central to the model's implementation. Key components include:

- Positional Encoding: The "PositionalEncoding" class injects positional information into input data to help the model understand sequential relationships.
- Residual Blocks: The "ResidualBlock" class defines a block with two convolutional layers, incorporating skip connections to alleviate vanishing gradient issues.
- Convolutional Neural Network (CNN): Represented by the "ConvNet" class.
- UNet Architecture: The "UNet" class, used for image segmentation, features symmetric downsampling and upsampling paths to capture features at different scales.

Configurations like "convnet_small_cfg" and "unet_res_cfg" tailor the network architecture to specific tasks. The "build_network" function instantiates the neural network based on the chosen configuration.

Training the DDPM model involves setting the diffusion simulation time step. The model outputs denoised images at various time steps. Figure 5 shows the model's output images at time steps 100, 500, 1000, and 1500, respectively.

By observing denoised images at different prediction time steps, a clear trend emerges: image clarity and recognizability increase with longer time steps. At a prediction time step of 100, the images are disorganized, hindering digit recognition. By 500 steps, some improvement is seen, but distortions persist. At 1000 and 1500 steps, the images show sharper digits and less distortion, indicating significant quality improvement. This demonstrates the crucial role of prediction time steps in enhancing denoised image quality, as longer steps allow more effective noise removal and better restoration of detail and structure. This analysis underscores the DDPM model's effective denoising and image generation capabilities for handwritten digits.

4.2 Conditional Diffusion Models Implementation

Conditional diffusion models enhance the original model by integrating additional labeling information, which allows for more precise image generation based on specified labels. This is achieved through the inclusion of a label embedding layer in the model architecture. Specifically, the code snippet below demonstrates how the label embedding is implemented:

```
if num\_classes is not None:
self.label\_emb = nn.Embedding(num\_classes, time\_dim)
```

An Embedding layer is defined in the code. During forward propagation, category labels are embedded into the positional encoding of the time dimension, enabling the model to generate images based on input category label information and improving image generation performance. Since the MNIST dataset consists of grayscale images, the CIFAR-10 dataset, which contains color images, was chosen for training the Conditional Diffusion model. The additive noise picture of the forward process is shown in Figure 6

The images generated after training the model for 1, 30, 60 and 100 times are as follows:

By observing the Figure 7, it can be concluded that: at 30 training iterations, specific shapes are difficult to distinguish in most categories, though trucks are more recognizable; by 60 training iterations, more categories begin to show clear shapes, especially horses and trucks; at 100 training iterations, the outlines of most categories become clearer, with airplanes and birds distinctly showing shape features. In summary, the diffusion model performs well in processing natural image datasets.

4.3 AC-GANs Implementation

AC-GANs improve the generator's ability to create realistic images with specific attributes by incorporating category labels into the model architecture. In AC-GANs, the generator's input includes random noise and category labels, guiding it to produce images belonging to specific categories. The discriminator, meanwhile, not only determines the authenticity of the images but also categorizes them, learning both image authenticity and category recognition. This dual function enhances the discriminator's accuracy.

In the code, the 'Generator' class takes a 100-dimensional random noise vector and a category label as input. The generator uses transposed convolutional layers ('ConvTranspose2d') to convert the input noise into a 3-channel, 64x64 image, embedding conditional information after multiplying the noise and labels.

The output is normalized by 'Tanh' activation function and the resulting image pixel values are in the range [-1, 1].

Accordingly, the input to the discriminator is a 3-channel, 64x64 sized image.

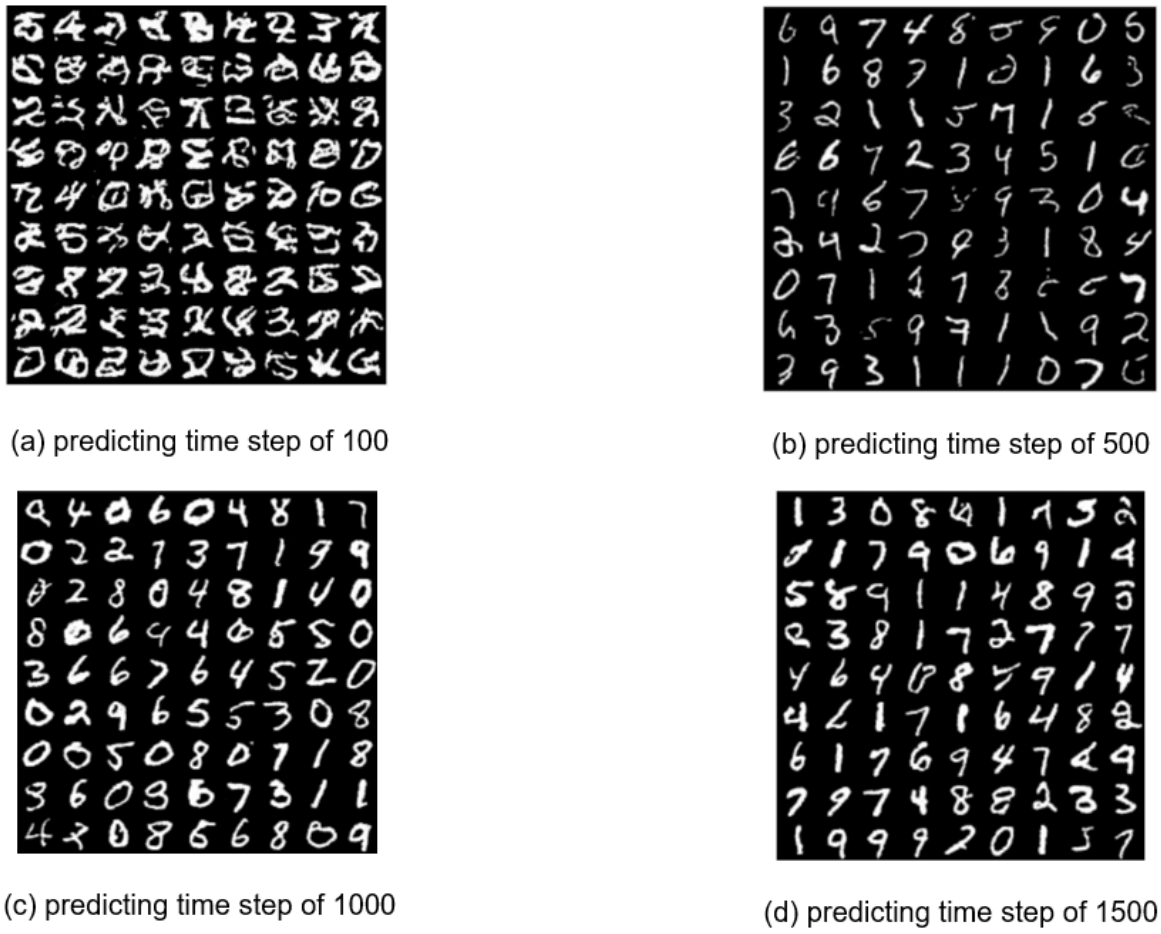


Figure 5: Denoised images from DDPM with different predicting time steps in MINIST

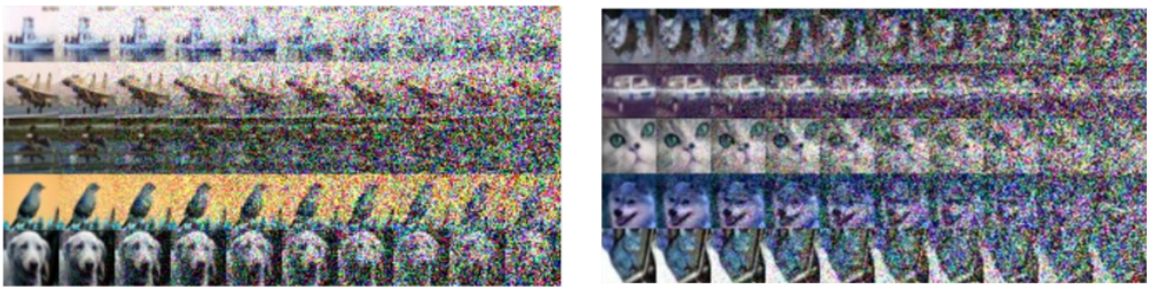


Figure 6: 10 random sets of noisy images with time steps 0-100.

The discriminator consists of a series of convolutional layers ('Conv2d'), a batch normalization layer ('BatchNorm2d'), a 'LeakyReLU' activation function, and a 'Dropout' operation for extracting features from the image.

The final output layer consists of two parts: a 'Sigmoid' activation function outputs the veracity score of the image and a 'LogSoftmax' layer outputs the category probability of the image. The specific category categorization code is as follows:

```
plabel = self.label\_layer(x)
plabel = plabel.view(-1,11)
```

The images generated after training the model for 1, 30, 60 and 100 times are shown in Figure 8

By observing the images generated after 30 epochs of training, it can be concluded that they exhibit a psychedelic effect, making it challenging to discern their basic shapes. At epoch 60, while the colors slightly resemble those of real images, the shapes remain

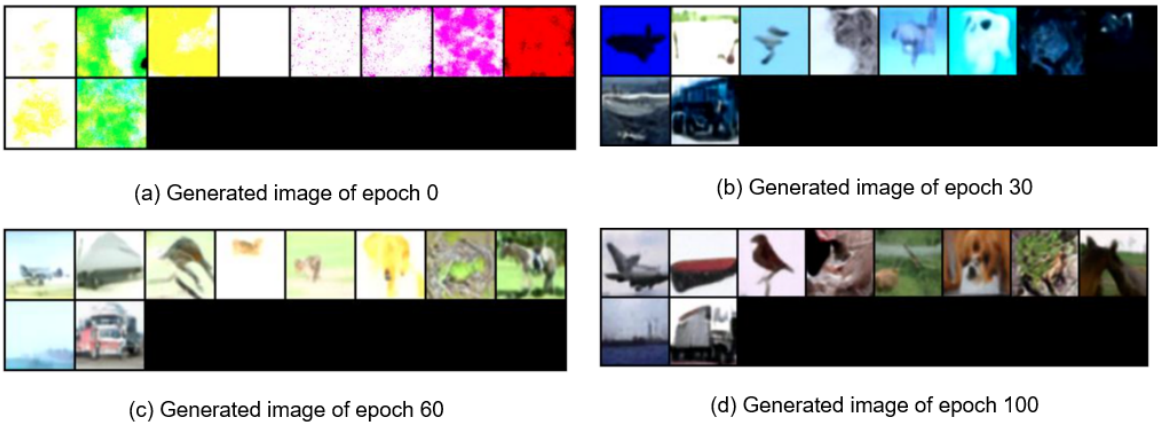


Figure 7: Generated images from Diffusion models of different epoch.

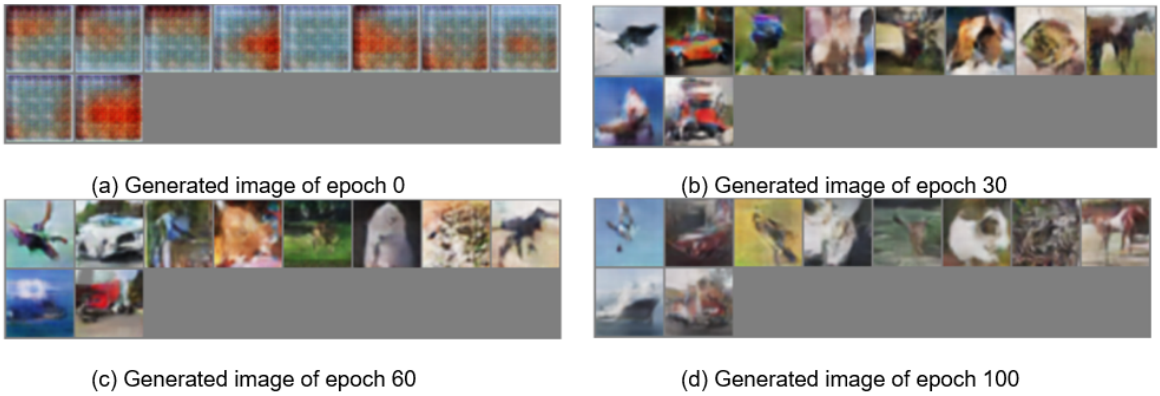


Figure 8: Generated images from AC-GANs of different epoch

difficult to discern. However, by epoch 100, the images display more normal colors, and most categories such as 'car', 'dog', 'horse', and 'ship' can be distinguished. In summary, AC-GANs also performs well in processing natural image datasets.

4.4 Performance Comparison and Analysis

4.4.1 Training time. During the experiment, diffusion models required an average of 400 seconds per iteration, highlighting their longer training times due to the need for multiple inference steps. Conversely, GANs demonstrated faster training, with each epoch taking about 40 seconds, showcasing their efficiency.

4.4.2 Training loss. As Figure 9 shows, both models were trained 100 times under the same conditions. The diffusion model's loss function exhibited an initial sharp decrease followed by stabilization, indicating effective learning and convergence. In contrast, GANs showed oscillating loss functions for both generators and discriminators due to their adversarial nature, with the generator's loss not yet stabilized within 100 iterations.

Overall, diffusion models displayed higher training stability with smoother loss curves, while GANs, despite their faster per-epoch training time, required more iterations and careful tuning to achieve convergence.

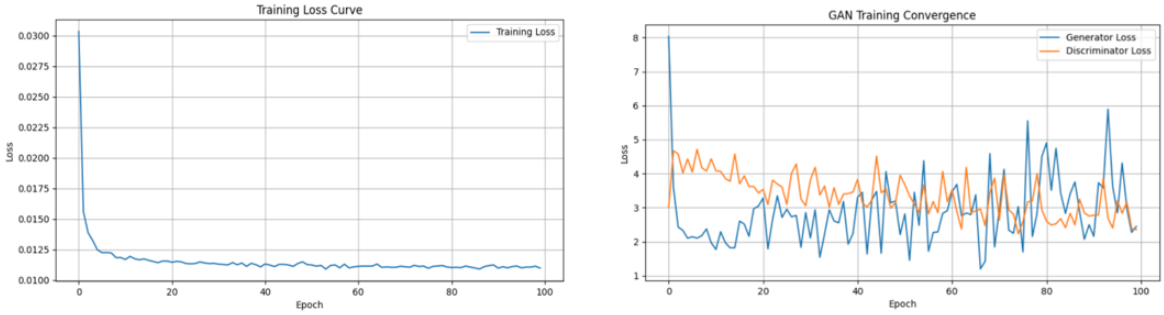
4.4.3 Generated Images Comparison. (1) Visual aspect:

By selecting two sets of image classes(Figure 10) where both diffusion models and GANs perform well, the following observations were made:

During the initial 0-30 epochs, GANs produce images with recognizable but fuzzy shapes, while diffusion models generate images with large color blocks lacking clear outlines. This indicates that GANs initially capture basic features but struggle with shape accuracy, resulting in slightly blurred images. Diffusion models, however, seem to prioritize color distributions over shapes early in training.

As training continues, GANs persist in generating images with blurred shapes and psychedelic colors, struggling to achieve clear contours and realistic colors. This ongoing issue with shape clarity and color fidelity suggests that GANs face challenges in fine image generation.

In contrast, the performance of diffusion models improves over time, producing images with sharper shapes and more realistic colors. With continued training, diffusion models enhance their



(a) Training loss curve of Diffusion model (b) Training loss curve of GANs

Figure 9: Training loss curves of different models

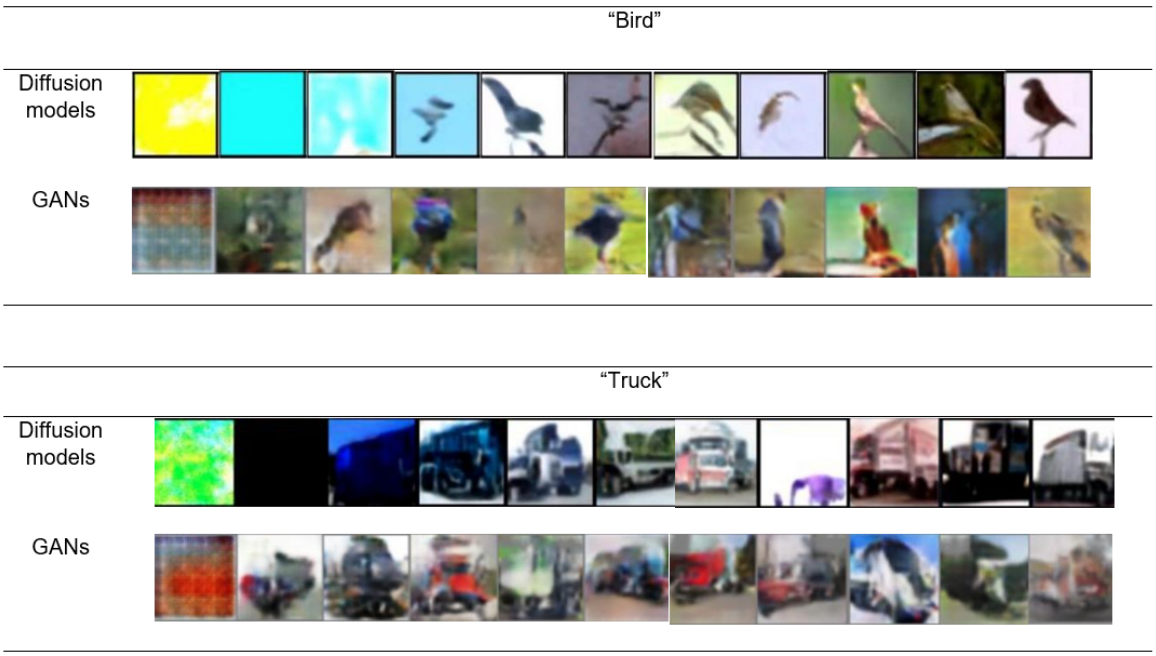


Figure 10: Two sets of image classes generated by Diffusion models and GANs

ability to capture complex details and color distributions, resulting in images that closely resemble real photographs. This superior performance becomes more apparent after 100 training sessions.

In summary, while GANs initially capture some visual features, they struggle with shape precision and color realism over extended training. Diffusion models, however, show continuous improvement, generating high-quality images with clear shapes and accurate colors, making them a promising alternative for image generation tasks.

(2) Numeric aspect:

Since visual assessment can be influenced by subjective factors, the Fréchet Inception Distance (FID) was introduced to increase the rigor of the assessment. FID employs the Inception network, trained for image classification on the ImageNet dataset, to extract features from both generated and real images. This allows FID to quantify the diversity of generated images and their similarity to real images.

Specifically, FID compares the distribution of 2,048-dimensional embedding vectors from the penultimate layer of the Inception-v3 network between generated and real images [33]. It calculates the mean and covariance matrix from the data samples to compute the Fréchet distance, a measure of similarity between two probability distributions. This provides an objective and reliable metric for assessing the performance of generative models. The formula for FID is shown in Equation(10):

$$FID\left(X,\ X'\right) = \mu_X - \mu_{X'2}^2 + Tr(\sum X + \sum X' - 2\sqrt{\sum X \sum X'} \tag{10}$$

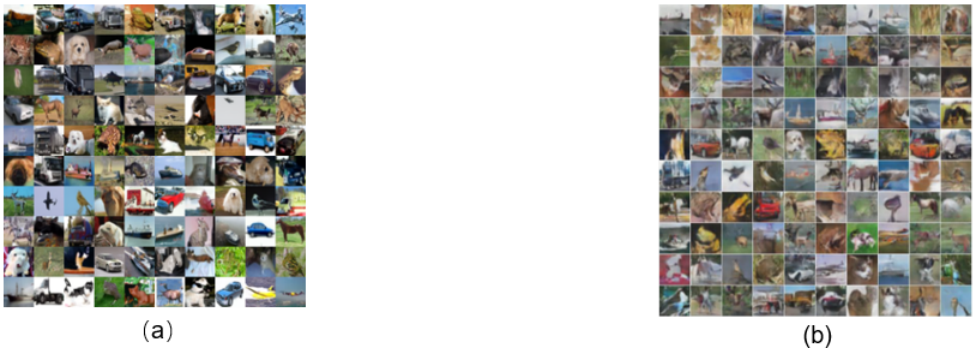


Figure 11: Generated image datasets from trained (a) Diffusion models and (b) GANs

Table 1: FID metric of Diffusion models and GANs

	FID
Diffusion models	68.3530181639046
GANs	76.6403253100458

First, two generative image datasets are generated using two already trained generative models, respectively. The generated images are shown as below:

Next, calculate the FID between the generated image dataset and the real dataset CIFAR-10 using the code shown below.

```
fid\_score\_train = calculate\_fid\_given\_paths([gan
\_images\_path, cifar10\_dir],
batch\_size=50, device='cpu', dims=2048)
print(f'FID score between GAN images and CIFAR-
10 set:
   {fid\_score\_train}')
fid\_score\_train = calculate\_fid\_given\_paths(
[Diffusion\_images\_path, cifar10\_dir],
batch\_size=50, device='cpu', dims=2048)
print(f'FID score between Diffusion images and CIFAR-
10
   set: {fid\_score\_train}')
```

The calculated FID metric is shown in the Table 1

From the obtained data, it is evident that the FID score between the image set generated by the diffusion model and the original image dataset is lower compared to that of the image set generated by GANs, with a difference of about 8. This indicates that the images generated by the diffusion model are more similar to real images, implying higher quality and accuracy compared to GANs.

In this study, three image generation models were implemented and evaluated: DDPM, conditional diffusion model, and AC-GAN. The code implementation and architecture of each model were thoroughly detailed. A comprehensive analysis of the generated images provided insights into the strengths and limitations of both GANs and diffusion models.

5 Conclusion

This report investigates image generation using MNIST and CIFAR-10 datasets, focusing on diffusion models and GANs. A detailed comparison of these models highlights their strengths and weaknesses. The results indicate that both diffusion models and GANs perform well in image generation. Diffusion models have a more stable training process and excel in later training stages, producing images with sharper shapes and more realistic colors. GANs, while faster in training, are less stable due to their adversarial nature and struggle with generating precise shapes and colors over time. In summary, this report provides insights into the performance of diffusion models and GANs, contributing to the ongoing development of more advanced image generation techniques.

Acknowledgments

This research was supported by "Chunhui Plan" Project for International Cooperative Scientific Research of Ministry of Education of China (HZKY20220134), Major Program of Natural Science Foundation for Jiangsu Higher Education Institutions of China (22KJA520001), Suzhou Science and Technology Development Planning Programme (SYG202316), Xi'an Jiaotong-Liverpool University (XJTLU) Research Development Fund (RDF-22-02-048) and Teaching Development Fund (TDF23/24-R27-232). This research utilized XJTLU's HPC facility, supported by XJTLU Research IT.

References

[1] R. Zhu, "Generative Adversarial Network and Score-Based Generative Model Comparison," 2023 IEEE International Conference on Image Processing and Computer Applications (ICIPCA), Changchun, China, 2023, pp. 1-5, doi: 10.1109/ICIPCA59209.2023.10258000.

[2] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion Models in Vision: A Survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 9, pp. 10850-10869, 1 Sept. 2023, doi: 10.1109/TPAMI.2023.3261988.

[3] D. M. Shariff Mohana, A. H and A. D, "Artificial (or) Fake Human Face Generator using Generative Adversarial Network (GAN) Machine Learning Model," 2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT), Erode, India, 2021, pp. 1-5, doi: 10.1109/ICECCT52121.2021.9616779.

[4] V. S. Krishna Katta, H. Kapalavai, and S. Mondal, "Generating New Human Faces and Improving the Quality of Images Using Generative Adversarial Networks(GAN)," 2023 2nd International Conference on Edge Computing and Applications (ICECAA), Namakkal, India, 2023, pp. 1647-1652, doi: 10.1109/ICECAA58104.2023.10212099.

[5] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," arXiv e-prints, 2017. doi:10.48550/arXiv.1703.10593.

[6] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 5967-5976, doi: 10.1109/CVPR.2017.632.

[7] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 8798-8807, doi: 10.1109/CVPR.2018.00917.

[8] T. Kim, M. Cha, H. Kim, *et al.*, "Learning to Discover Cross-Domain Relations with Generative Adversarial Networks," arXiv e-prints, 2017.

[9] C. Ledig, L. Theis, F. Huszár, *et al.*, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," 2017. [Online]. Available: http://arxiv.org/abs/1609.04802.

[10] B. Wu, H. Duan, Z. Liu, *et al.*, "SRPGAN: Perceptual Generative Adversarial Network for Single Image Super Resolution," 2017. DOI:10.48550/arXiv.1712.05927.

[11] X. Wang, K. Yu, S. Wu, *et al.*, "ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks," 2018. DOI:10.1007/978-3-030-11021-5_5.

[12] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 8798-8807, doi: 10.1109/CVPR.2018.00917.

[13] A. Antoniou, A. Storkey, and H. Edwards, "Data Augmentation Generative Adversarial Networks," arXiv e-prints, 2017. doi:10.48550/arXiv.1711.04340.

[14] V. Gupta, R. Sadana, and S. Moudgil, "Image style transfer using convolutional neural networks based on transfer learning," International journal of computational systems engineering, 2019, 5(1), 53-60. DOI:10.1504/IJCSYSE.2019.10019689.

[15] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are GANs created equal? A large-scale study," in Advances in neural information processing systems, pp. 700-709, 2018.

[16] Y. Li, Q. Wang, J. Zhang, L. Hu, and W. Ouyang, "The theoretical research of generative adversarial networks: an overview," Neurocomputing, vol. 435, pp. 26–41, 2021. https://doi.org/10.1016/j.neucom.2020.12.114.

[17] R. Soleymanzadeh and R. Kashef, "The Analysis of the Generator Architectures and Loss Functions in Improving the Stability of GANs Training towards Efficient Intrusion Detection," 2022 9th International Conference on Soft Computing & Machine Intelligence (ISCMI), Toronto, ON, Canada, 2022, pp. 246-252, doi: 10.1109/ISCMI56532.2022.10068468.

[18] H. Thanh-Tung and T. Tran, "Catastrophic forgetting and mode collapse in GANs," 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 2020, pp. 1-10, doi: 10.1109/IJCNN48605.2020.9207181.

[19] H.-Y. Chen and C.-J. Lu, "Nested Variance Estimating VAE/GAN for Face Generation," 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 2019, pp. 1-8, doi: 10.1109/IJCNN.2019.8852154.

[20] P. K. Saluja and D. Vathana, "Rectifying Mode Collapse in GANs," 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2022, pp. 1718-1722, doi: 10.1109/ICAAIC53929.2022.9792861.

[21] A. Razavi, A. van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with VQ-VAE-2," arXiv:1906.00446, 2019.

[22] C. Nash, J. Menick, S. Dieleman, and P. W. Battaglia, "Generating images with sparse representations," arXiv:2103.03841, 2021.

[23] R. Child, "Very deep vaes generalize autoregressive models and can outperform them on images," arXiv:2011.10650, 2021.

[24] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," arXiv:1503.03585, 2015.

[25] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," arXiv:1907.05600, 2020.

[26] A. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," arXiv:2102.09672, 2021.

[27] X. Li, "Diffusion Models for Image Restoration and Enhancement – A Comprehensive Survey," arXiv e-prints, 2023. doi:10.48550/arXiv.2308.09388.

[28] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," Int Conf Mach Learn, pp. 2256-2265, 2015.

[29] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," Adv Neural Inf Process Syst, vol. 33, pp. 6840-6851, 2020.

[30] O. Dalmaz, B. Saglam, G. Elmas, M. Mirza, and T. Çukur, "Denoising Diffusion Adversarial Models for Unconditional Medical Image Generation," 2023 31st Signal Processing and Communications Applications Conference (SIU), Istanbul, Turkiye, 2023, pp. 1-5, doi: 10.1109/SIU59756.2023.10223912.

[31] P. Dhariwal, A. Nichol, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, *et al.*, "Diffusion models beat GANs on image synthesis," Advances in Neural Information Processing Systems, vol. 34, pp. 8780-8794, 2021.

[32] J. Lee and M. Lee, "FIDGAN: A Generative Adversarial Network with An Inception Distance," 2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Bali, Indonesia, 2023, pp. 397-400, doi: 10.1109/ICAIIC57133.2023.10066964.

[33] lherranz. (2018, August 7). Generative adversarial networks and image-to-image translation [Online]. Available: http://www.lherranz.org/2018/08/07/imagetranslation/

[34] DeepHub IMBA. (2023, January 10). DeepHub IMBA [Online]. Available: https://mp.weixin.qq.com/s/DnOIQIdL8JKi0WZjiL_n2A

[35] xirongxu_dlut. (2023, September 14). In-depth Diffusion Model (Diffusion Model) Series: Cornerstone DDPM (Model Architecture), the most detailed DDPM architecture diagrams [Online]. Available: https://blog.csdn.net/xirongxu_dlut/article/details/132873922

[36] J. Song, C. Meng, and S. Ermon, "Denoising Diffusion Implicit Models," 2020. DOI: 10.48550/arXiv.2010.02502.

[37] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," Proc. Int. Conf. Learn. Representations, 2021.

[38] C.-W. Huang, J. H. Lim, and A. C. Courville, "A variational perspective on diffusion-based generative models and score matching," Proc. Int. Conf. Neural Inf. Process. Syst., pp. 22863-22876, 2021.

[39] C. Saharia, J. Ho, W. Chan, *et al.*, "Image Super-Resolution via Iterative Refinement," 2021. DOI: 10.48550/arXiv.2104.07636.