Proceedings of CONF-SPML 2026 Symposium: The 2nd Neural Computing and Applications Workshop 2025
DOI: 10.54254/2755-2721/2026.TJ28950

Diffusion Models for Image Generation Acceleration Techniques and Advances

Zihao Cai[1], Qingyu Li[2], Jin Liu[3*]

[1]*School of Information and Intelligent Engineering, Guangzhou Xinhua University, Dongguan, China*
[2]*School of Computer Science and Technology, Chongqing College of Mobile Communication, Chongqing, China*
[3]*School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang,China*
**Corresponding Author. Email: 3210904022@stmail.ujs.edu.cn*

Abstract. Nowadays, diffusion models have been widely used in generative tasks, including image generation, owing to their advantages in generation quality and stability. However, the heavy computational burden and slow generation speed of diffusion models have, to a certain extent, restricted their application and widespread adoption. To address this issue, various methods have been developed to accelerate the image generation task based on diffusion models. This paper systematically reviews these acceleration methods and categorizes them into four categories based on different dimensions: sampling strategy optimization, latent space diffusion, model architecture innovation, and training strategy optimization. Regarding sampling strategy optimization, this paper introduces several mainstream sampling methods based on their developmental trajectory. Regarding latent space diffusion, this paper mainly introduces its principles and workflow. Regarding model architecture innovation, this paper focuses on DiT and DiM. Regarding training strategy optimization, this paper focuses on progressive distillation and consistency models. Furthermore, this paper summarizes these acceleration methods and proposes future research directions.

Keywords: Diffusion Models, Image Generation, Acceleration Techniques, Efficient Sampling

1. Introduction

The current denoising diffusion model has surpassed traditional methods such as GANs in performance and reached the industry-leading level due to its excellent quality and extremely high generation diversity in image generation tasks [1]. Diffusion models have been widely applied to a variety of generative modeling tasks, and image generation is one of the typical application scenarios [2]. However, traditional diffusion models typically require hundreds to thousands of iterations to produce a single valid output. This inefficiency is particularly problematic when scaling to high-dimensional data or deploying models in lightweight applications. Furthermore, training

© 2025 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.TJ28950

diffusion models typically requires significant computational resources [3]. These problems limit further extensive research and application of diffusion models.

To overcome this critical bottleneck, academia, and industry have developed a variety of acceleration techniques for diffusion model image generation. These acceleration methods have significantly improved the efficiency and practicality of diffusion models. However, a systematic review of these acceleration methods is lacking. This makes it difficult for researchers and engineers to quickly grasp the overall landscape of the field, understand the connections and differences between different acceleration methods, and select the optimal acceleration solution for specific scenarios.

Therefore, this article introduces the fundamental principles of the Diffusion Model and discusses several key challenges it faces in image generation tasks. It then categorizes and introduces mainstream diffusion model acceleration methods by technical paradigm. Finally, it provides an outlook on future developments in this field and provides a summary. The purpose of this paper is to help engineers and researchers make sound technical decisions based on particular application requirements.

2. Preliminary

A denoising diffusion probabilistic model (DDPM) uses two Markov chains, one forward chain that converts the original data into noise, and the other reverse chain that restores the noise back to the data. The forward chain converts the data distribution into an uncomplicated prior distribution, and the reverse chain reverses this process. Initially, a random vector is drawn from the prior distribution, after which new data points are produced through ancestral sampling utilizing the reverse Markov chain [1].

Although diffusion models have made considerable strides in image generation tasks, their fundamental paradigm presents several significant challenges that need to be addressed. The first is slow sampling. The denoising process requires the model to perform hundreds or even thousands of full forward passes to produce a single image. Each pass requires computation on a massive neural network (such as a U-Net), resulting in extremely high generation latency, often measured in seconds or even minutes. This severely limits its application in scenarios such as real-time interaction, video generation, and large-scale deployment. Another issue is the significant consumption of computational resources. The slow sampling rate forces image generation tasks to occupy high-performance computing units for extended periods, resulting in high energy consumption and financial costs. This not only increases operational pressure on service providers but also makes running high-quality diffusion models impractical on resource-constrained devices.

Therefore, one approach to optimization is to decrease the overall number of sampling steps to accelerate the generation process. Nonetheless, because the reverse process is a discretized estimate of a continuous stochastic differential equation (SDE), drastically lowering the number of steps will enhance the discretization error, frequently resulting in a noticeable decline in the quality of the produced image, which may manifest as artifacts, blurriness, or a reduction in detail. Therefore, how to decrease the number of sampling steps while maintaining generation quality, constitutes a core challenge in acceleration technology research.

In summary, addressing the issues of slow sampling and high computational costs is key to moving diffusion models from academic research to widespread practical application. To address these challenges, academia and industry have developed a wide variety of acceleration technologies, which will be detailed later in this article.

DOI: 10.54254/2755-2721/2026.TJ28950

3. Sampling strategy optimization

Sampling methods are essential in the process of image generation for diffusion models. Specifically, they guide the model in gradually restoring a clear image from pure noise, effectively achieving the reverse evolution from random noise perturbations to the target distribution. Different sampling strategies directly impact the quality, detail fidelity, and speed of the generated images.

By optimizing the sampling method of the diffusion model, higher-quality images can be generated in a shorter time and with fewer iterations.

Sampling acceleration of diffusion models has undergone a gradual evolution. The initial DDPM were relatively slow in sampling efficiency, and the subsequently proposed Denoising Diffusion Implicit Models (DDIM) improved the generation speed to a certain extent. During this period, there were also studies that formalized the diffusion process as a problem of solving Ordinary differential equations (ODEs), thereby introducing classical numerical solvers (such as Euler, Heun, LMS, etc.) to approximate and achieve more efficient sampling. However, traditional ODE methods still have limitations in stability and efficiency, which has prompted researchers to design dedicated efficient sampling algorithms based on the characteristics of diffusion models. This article focuses on reviewing the representative efficient sampling methods proposed in recent years for accelerating diffusion models.

3.1. DPM-Solver

DPM-Solver is an efficient ODE solver designed specifically for diffusion models. Experimental results by Lu et al. show that DPM-Solver can achieve image generation results that match or even surpass those of the DDIM method, which requires only 10 iterations, in image generation tasks [4].

Lu et al. significantly improved the effectiveness of the sampler in their DPM-Solver study, and were able to generate high-quality samples with less than 10 number of function evaluations (NFEs). Lu et al. introduced multiple variations of DPM-Solver, which include various order types. Additionally, they suggested a scheme for adaptive step-size scheduling for their sampler. Overall, DPM-Solver can be utilized for both continuous-time and discrete-time DPM, as well as to conditional sampling with classifier guidance [4].

In summary, compared with the classic DDIM sampling, DPM-Solver can generate higher-quality images using a much lower number of sampling steps, and the image generation speed is greatly improved.

3.2. DPM-Solver++

DPM-Solver++ is an improved sampler that can utilize the threshold method directly to enhance the stability of the sampling process. Lu et al. proposed an improved sampler, DPM-Solver++, as well as its single-step and multi-step variants.

Experimental findings indicate that DPM-Solver++ is capable of producing samples of better quality compared to DPM-Solver within 15-20 NFEs, and it can be applied to both pixel space and latent space DPM [5].

DOI: 10.54254/2755-2721/2026.TJ28950

Table 1. Sample quality of different samplers measured by FID ↓ , varying the NFE[5]

Guidancescale	Thresholding	Samplingmethod	NFE						
			10	15	20	25	50	100	250
8.0	No	DDIM	13.04	11.27	10.21	9.87	9.82	9.52	9.37
		DPM-Solver-2	114.62	44.05	20.33	9.84	-	-	-
		DPM-Solver-3	164.74	91.59	64.11	29.40	-	-	-
		DPM-Solver++(M)	14.44	9.46	9.10	9.11	-	-	-
	Yes	DDIM	10.58	9.53	9.12	8.94	8.58	8.49	8.48
		DPM-Solver++(S)	9.26	8.93	8.40	8.63	-	-	-
		DPM-Solver++(M)	9.56	8.64	8.50	8.39	-	-	-

Table 1 lists the FID of different sampling methods at different function iterations (ImageNet 256×256 dataset) with a guidance scale of 8.0 and with or without thresholding enabled. The results show that within the NFE range of 10–25, DPM-Solver++ produces images of superior quality to the other samplers in Table 1 for the same NFE.

In general, compared with DPM-Solver, the improved DPM-Solver++ sampler can generate higher quality images under the same NFE, especially performing well at a lower NFE of 10-20, which improves the efficiency of image generation.

3.3. UniPC

Zhao et al. introduced a framework known as UniPC for fast sampling of discrete-time variable models (DPMs) without training. UniPC significantly outperforms existing methods within 5-10 NFEs and can achieve better sampling quality with an increased quantity of sampling steps. Specifically, Zhao et al. developed a unified corrector (UniC) that employs the model output $\epsilon(x, t)$ at the present time step t to derive the precise x. Unlike the predictor-corrector paradigm in numerical differential equation solvers that requires double NFE, UniC reuses the model output $\epsilon(x, t)$ for the next sampling step, thus eliminating the need for additional function evaluations. UniC can be utilized following any current DPM sampler to enhance the order of accuracy with minimal impact on the inference speed [6].

Experiments show that UniPC performs significantly better than existing methods with 5-10 NFEs and can achieve better sampling quality with more sampling steps. Overall, UniPC shows excellent performance when sampling pre-trained DPMs unconditionally and conditionally in both pixel space and latent space [6]. The UniPC sampler generates images more efficiently than DPM-Solver++ at very low generation steps (5-10 NFE).

3.4. DPM-Solver-v3

Zheng et al. proposed a fast solver called DPM-Solver-v3, which includes a multi-step prediction-correction framework of arbitrary order and uses various techniques to improve sampling, thereby achieving low-step sampling or large-scale sampling. Research shows that DPM-Solver-v3 is effective for unconditional conditional sampling of pre-trained DPMs in both pixel space and latent space, and the sample quality is significantly improved within 5-10 steps [7].

DOI: 10.54254/2755-2721/2026.TJ28950

Table 2. Quantitative results on CIFAR10, measured by FID ↓ [7]

Method	Model	NFE							
		5	6	8	10	12	15	20	25
DPM-Solver++	ScoreSDE	28.53	13.48	5.34	4.01	4.04	3.32	2.90	2.76
UniPC		23.71	10.41	5.16	3.93	3.88	3.05	2.73	2.65
DPM-Solver-v3		12.76	7.40	3.94	3.40	3.24	2.91	2.71	2.64
DPM-Solver++	EDM	24.54	11.85	4.36	2.91	2.45	2.17	2.05	2.02
UniPC		23.52	11.10	3.86	2.85	2.38	2.08	2.01	2.00
DPM-Solver-v3		12.21	8.56	3.50	2.51	2.24	2.10	2.01	2.00

Table 2 shows the FID of several mainstream samplers under different diffusion model frameworks (ScoreSDE and EDM) at different NFEs, comparing their performance in image generation quality. Experimental results show that, under both ScoreSDE and EDM models, when NFE is 5-25, DPM-Solver-v3 achieves lower FID in most cases, indicating better image generation quality.

In general, DPM-Solver-v3 can generate images with better detail quality than UniPC at lower NFE (especially NFE of 5-10), which improves the efficiency of image generation.

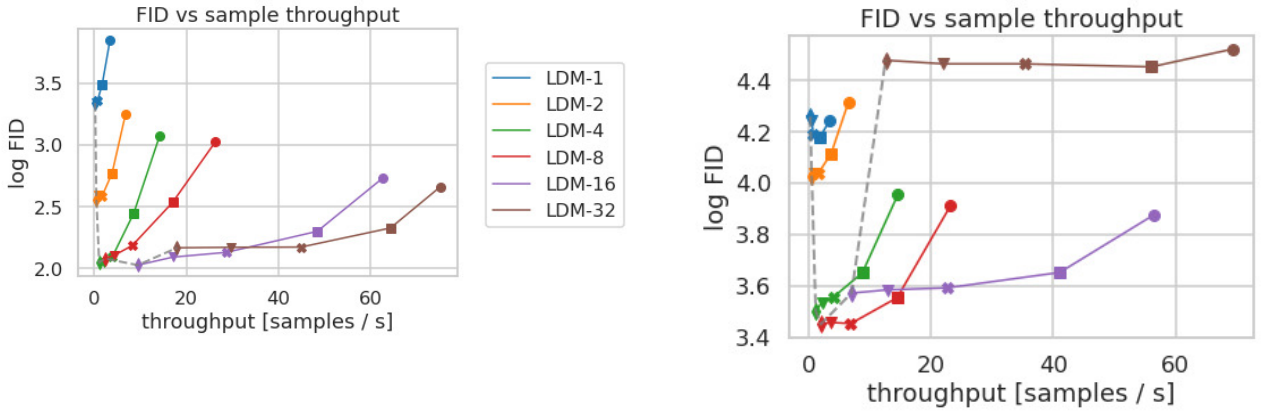
4. Latent space diffusion

Latent diffusion model (LDM) compresses a high-resolution image into a much smaller summary that retains the core semantic information, known as a latent representation. The diffusion model performs the entire process of denoising and adding noise in this much smaller latent space. Finally, a decompressor is used to restore the denoised latent representation to a high-resolution image, significantly improving the speed of image generation.

In LDM, the compression factor is an important concept, which is the ratio between the original image's spatial resolution and the latent space resolution. The higher the compression factor, the smaller the size of the corresponding latent space and the faster the image generation speed.

Specifically, assuming the original image resolution is H×W, the autoencoder compresses it to h×w, then the compression factor is:

$$compression\ factor = \frac{H}{h} = \frac{W}{w} \tag{1}$$



DOI: 10.54254/2755-2721/2026.TJ28950

Figure 1. The speed of inference and the quality of samples, assessed by $log^{FID}\downarrow$, are evaluated with varying degrees of compression on the CelebA-HQ (left) and ImageNet (right) datasets [8]

Figure 1 shows the relationship between compression factor and inference speed, and generation quality for generation tasks. The horizontal axis of the chart represents throughput, or the speed at which samples are generated. The vertical axis is $\log^{\mathrm{FID}}$, represents generation quality. Different line segments represent LDMs with different compression factors.

Experiments show that the compression factor determines the dimensionality of the latent space operated by the diffusion process, which directly affects the computational complexity of the model. On the one hand, when the compression factor rises, the dimensionality of the latent space is notably diminished, resulting in a reduction in the computational cost of a single denoising step, which in turn allows for higher throughput. On the other hand, overly aggressive compression (such as LDM-16 and LDM-32 in Figure 1) creates a bottleneck that limits the model's ability to reconstruct high-fidelity images from the latent representation, thereby setting an upper limit on the best achievable image quality (measured by the lowest FID).

In the Latent Diffusion Model (LDM), there exists an optimal compression factor that significantly increases model generation speed while maintaining or even exceeding the image generation quality of the original uncompressed model. Therefore, when using LDM for image generation tasks, simply using an appropriate compression factor can significantly increase image generation speed with little impact on image quality.

5. Model architecture innovation

The model architecture reform is to replace the U-Net model that denoises the latent space with models of other architectures to achieve higher image generation efficiency.

5.1. DiT & Δ-DiT

Peebles et al. proposed Diffusion Transformers (DiTs). This represents a novel diffusion model that utilizes the Transformer architecture. Instead of the typical U-Net backbone, they developed a latent diffusion model trained on images, employing a Transformer that works on latent patches[9]. DiTs exceeded the performance of the earlier U-Net model and inherited the remarkable scalability typical of the Transformer model category.

Table 3. Performance comparison of various methods, including DiT, on the ImageNet 256×256 dataset[9]

Model	FID↓	sFID↓	IS↑	Precision↑	Recall↑
StyleGAN-XL	2.30	4.02	265.12	0.78	0.53
LDM-8-G	7.76	-	209.52	0.84	0.35
LDM-4-G (cfg=1.25)	3.95	-	178.22	0.81	0.55
LDM-4-G (cfg=1.50)	3.60	-	247.67	0.87	0.48
DiT-XL/2	9.62	6.85	121.50	0.67	0.67
DiT-XL/2-G (cfg=1.25)	3.22	5.28	201.77	0.76	0.62
DiT-XL/2-G (cfg=1.50)	2.27	4.60	278.24	0.83	0.57

DOI: 10.54254/2755-2721/2026.TJ28950

Table 3 provides a quantitative comparison of the performance of DiT against other leading models, including five dimensions: FID, sFID, IS, Precision, and Recall.

Experimental results show that compared with the LDM method and StyleGANXL method, DiT-XL exhibits the highest generation quality among all previous models, with high generation quality and high scalability.

Nevertheless, because of the intricate nature of the Transformer architecture, the generation speed of DiT is relatively slow compared to the traditional U-Net model. To solve this problem, Chen et al. proposed Δ-DiT, a method for acceleration that does not require training and is specifically designed for DiTs. [10].

Chen et al. proposed a training-free caching mechanism, Δ-Cache, specifically for DiT, which improves the image generation speed of the DiT method while maintaining the same quality. According to their experimental results, Δ-DiT makes DiT 1.60 times faster in image generation tasks overall [10].

5.2. DiM

DiM (Diffusion Mamba) proposed by Teng et al. replaces the backbone network of the diffusion model from U-Net to the sequence model Mamba based on the state space model (SSM)[11].

Table 4. Pretraining on ImageNet with the resolution 256 × 256[11]

Mode	Images (Iterations × BatchSize)	Parameters	GFlops	FID↓
DiT	1792M (7M × 256)	675M	119	2.27
DiM-Large	300M (300K × 1024)	380M	94	2.64
DiM-Huge	319M (425K × 768)	860M	210	2.40
DiM-Huge	480M (625K × 768)	860M	210	2.21

Table 4 compares the amount of training data (Images), model size (Parameters), computational cost (GFlops), and image generation quality (FID) of DiT and DiM.

Experiments show that the DiT model, trained with 675M parameters and 1792M images, achieved an FID score of 2.27. In contrast, the DiM-Huge model, which has a larger number of parameters (860M) and a higher computational cost (210 GFlops), achieved an FID score of 2.21 after training with only 480M images (far fewer than DiT), surpassing the baseline model DiT. This demonstrates that while the DiM-Huge model architecture requires more computational power, it is more data efficient and can achieve better image generation quality than DiT with significantly less training data.

In general, DiM delivers performance that is on par with other Transformer-based diffusion models for generating high-resolution images [11]. Furthermore, because the computational complexity of Mamba scales linearly with the number of tokens, while the computational expense of Transformers increases quadratically in relation to the number of tokens, DiM is particularly suitable for efficient high-resolution image generation due to its lower computational cost.

6. Training strategy optimization

Optimize model efficiency at the model training level to enhance the effectiveness of the diffusion model, which involves training strategy optimization. The current mainstream methods include Progressive distillation (PD) and Consistency distillation (CD).

DOI: 10.54254/2755-2721/2026.TJ28950

6.1. Progressive distillation

Progressive Distillation involves taking a pre-trained teacher model that requires many steps to generate images and teaching a student model to achieve the same quality in half the number of steps. This process can be repeated incrementally until the number of steps is reduced to a very small number. This enables the trained model to produce images of the same quality with fewer samples, ultimately increasing the speed of image generation.

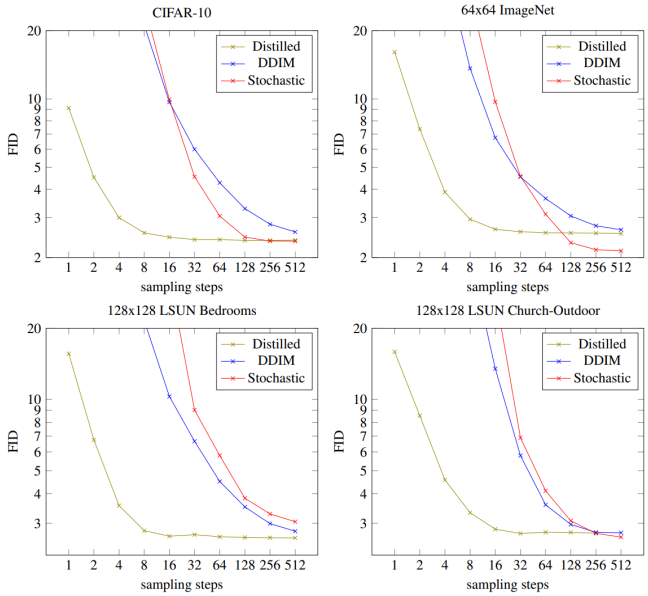


Figure 2. Sample quality results as measured by FID↓ [12]

Figure 2 compares the image generation quality of the DDIM sampler, optimized stochastic sampler, and the distilled model at different sampling steps, including four different test datasets.

Experimental findings indicate that the PD method is capable of producing images of superior quality with fewer iterations, which significantly enhances the efficiency of image generation.

6.2. Consistency models / consistency distillation

The Consistency Models proposed by Song et al. train a function that can directly map the noisy image x_t at any moment back to the final clear image x_0 in one step [13].



(a) CIFAR-10 (b) ImageNet 64x64 (c) Bedroom 256×256 (d) Cat 256×256

Figure 3. Comparison of image generation performance between consistency distillation (CD) and PD, measured by Euclidean distance↓ and LPIPS↓N [13]

Figure 3 compares the image generation quality of PD and Consistency Distillation at 1-4 sampling steps. Experimental results show that CD outperforms PD in most cases.

DOI: 10.54254/2755-2721/2026.TJ28950

Consistency models are based on a consistency property, which states that all points along the same diffusion trajectory (noisy images with varying degrees of noise) should be mapped to the same result (the original clean image). Consistency Models enable high-quality single-step or very few-step generation with extreme flexibility. Single-step generation for maximum speed is possible, while multi-step sampling, like with conventional diffusion models, is also possible. This multi-step consistency sampling approach often yields very good results, sacrificing some speed in exchange for higher generation quality.

Multi-step image generation using CD is generally more efficient than PD. This groundbreaking research has pushed the trade-off between speed and quality of generative models to a new level, greatly broadening the application potential of diffusion models in real-time or low-latency scenarios.

7. Conclusion

Although existing acceleration technologies have significantly shortened inference time by reducing NFEs and optimizing the generation process, the diffusion model still has high requirements for hardware performance in actual deployment. The parameter scale and video memory usage of the model have not been substantially reduced, and some architectural innovations have even brought additional computational overhead (such as DiT). On terminal devices with limited resources (like mobile devices and embedded systems), these methods still find it difficult to achieve real-time inference while ensuring generation quality. At the same time, there is currently a lack of systematic integration with compression technologies such as model pruning, quantization, and sparsification, which makes it difficult to break through the minimum deployment threshold. Future research should focus more on maintaining generation quality while achieving efficient deployment of diffusion models on a wider range of hardware platforms through multi-dimensional model compression and collaborative optimization.

This article comprehensively reviews the diffusion model acceleration methods for image generation tasks from multiple perspectives. First, this article briefly introduces the basic principles of the denoising diffusion model and proposes the challenges encountered by diffusion models in image acceleration tasks and general solutions. Next, this article constructs a four-layer analytical framework, categorizing and introducing various acceleration methods along four dimensions. From optimizing sampling strategies that maintain the model, to reducing the computational dimensionality through LDM, further exploring LDM model architecture innovations, and finally discussing training strategy optimization that changes the training objective, this article strives to present a clear and comprehensive technical landscape. Through a horizontal comparison and analysis of various technologies, this article reveals their different trade-offs in speed, quality, flexibility, and implementation cost.

Authors contribution

All the authors contributed equally and their names were listed in alphabetical order.

References

[1] Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., ... & Yang, M. H. (2023). Diffusion models: A comprehensive survey of methods and applications. ACM computing surveys, 56(4), 1-39.

[2] Croitoru, F. A., Hondru, V., Ionescu, R. T., & Shah, M. (2023). Diffusion models in vision: A survey. IEEE transactions on pattern analysis and machine intelligence, 45(9), 10850-10869.

DOI: 10.54254/2755-2721/2026.TJ28950

[3] Abul-Fazl, S. D., Dina, R., & Fairuza, H. (2025). Diffusion Models at Scale: Techniques, Applications, and Challenges.
[4] Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., & Zhu, J. (2022). Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. Advances in neural information processing systems, 35, 5775-5787.
[5] Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., & Zhu, J. (2025). Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. Machine Intelligence Research, 1-22.
[6] Zhao, W., Bai, L., Rao, Y., Zhou, J., & Lu, J. (2023). Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. Advances in Neural Information Processing Systems, 36, 49842-49869.
[7] Zheng, K., Lu, C., Chen, J., & Zhu, J. (2023). Dpm-solver-v3: Improved diffusion ode solver with empirical model statistics. Advances in Neural Information Processing Systems, 36, 55502-55542.
[8] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10684-10695).
[9] Peebles, W., & Xie, S. (2023). Scalable diffusion models with transformers. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 4195-4205).
[10] Chen, P., Shen, M., Ye, P., Cao, J., Tu, C., Bouganis, C. S., ... & Chen, T. (2024). Δ-DiT: A Training-Free Acceleration Method Tailored for Diffusion Transformers. arXiv preprint arXiv: 2406.01125.
[11] Teng, Y., Wu, Y., Shi, H., Ning, X., Dai, G., Wang, Y., ... & Liu, X. (2024). Dim: Diffusion mamba for efficient high-resolution image synthesis. arXiv preprint arXiv: 2405.14224.
[12] Salimans, T., & Ho, J. (2022). Progressive distillation for fast sampling of diffusion models. arXiv preprint arXiv: 2202.00512.
[13] Song, Y., Dhariwal, P., Chen, M., & Sutskever, I. (2023). Consistency models.