Proceedings of CONF-MLA 2026 Symposium: Intelligent Systems and Automation: AI Models, IoT, and Robotic Algorithms
DOI: 10.54254/2755-2721/2026.LD36928

Efficient Diffusion Models for Image Generation: A Cost-Decomposition Perspective

Junyu Mai

School of Automation, Guangdong University of Technology, Guangzhou, China
13728811832@163.com

Abstract. Diffusion models have become a dominant approach to image generation, but their iterative denoising process still creates high inference latency, memory consumption, and deployment cost. Existing surveys summarize efficient diffusion models from the viewpoints of sampling, compression, architecture, or applications, but they often do not make explicit which part of the inference cost is reduced by each method. This short survey reorganizes efficient image diffusion methods through an inference cost-decomposition perspective. The overall inference cost is described by three terms: the number of denoising function evaluations, the cost of each denoising step, and auxiliary overhead from conditioning, decoding, memory movement, and deployment frameworks. Based on this view, representative methods are grouped into sampling-step reduction, few-step student models, per-step computation reduction, and system-level deployment optimization. The discussion highlights that efficient diffusion is not simply a matter of using fewer sampling steps. Practical acceleration requires balancing image quality, latency, memory, training cost, and hardware constraints. This paper provides a compact framework for comparing efficient image diffusion methods and for selecting acceleration strategies under different deployment scenarios.

Keywords: diffusion models, image generation, efficient inference, cost decomposition, model compression

1. Introduction

Diffusion models have become a central technical route for image generation because they combine high sample quality, stable optimization, and strong scalability. Early diffusion generative modeling introduced the idea of gradually corrupting data and learning a reverse generation process [1]. Score-based generative modeling further connected generation with gradients of the data distribution [2], while denoising diffusion probabilistic models (DDPM) established a widely used training and sampling framework [3]. Later developments such as DDIM, score-based stochastic differential equations (Score-SDE), and latent diffusion models (LDM) made diffusion models more flexible and more suitable for high-resolution image synthesis [4-6].

Despite these advantages, inference efficiency remains a major obstacle. Standard diffusion models generate an image through repeated denoising-network evaluations. In text-to-image systems, this denoising loop is also accompanied by text encoding, classifier-free guidance (CFG),

© 2026 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.LD36928

scheduler updates, latent decoding, memory access, and hardware-specific runtime overhead [7]. Therefore, the apparent sampling step count is only one part of the real cost. A method that reduces the number of steps may still be slow if each step uses a large denoising backbone, while a compressed model may still be inefficient if it requires many steps or has high auxiliary overhead.

Recent work has proposed many acceleration strategies. DDIM changes the sampling trajectory and reduces the number of denoising steps without retraining [4]. Progressive distillation transfers multi-step teacher behavior into a few-step student model [8]. LDM reduces the feature resolution processed during repeated denoising by moving the diffusion process into latent space [6]. Efficient diffusion surveys have organized such work from broad viewpoints including training, sampling, architecture, compression, and deployment [9]. However, for image generation systems, it is still useful to ask a more direct question: which part of the inference cost is reduced by a given method?

This paper addresses that question through a cost-decomposition perspective. Instead of treating efficient diffusion methods as isolated algorithmic categories, the paper interprets them as attempts to reduce one or more cost terms in the inference pipeline. The contribution is threefold. First, it presents a compact cost view for image diffusion inference. Second, it uses this view to summarize representative efficient diffusion methods. Third, it discusses the trade-offs that matter when choosing methods for practical deployment, including quality, latency, memory, training burden, and hardware dependence.

2. Cost-decomposition view of diffusion inference

For image generation, a diffusion model can be understood as a learned reverse denoising process that transforms noise into an image through multiple denoising steps [3]. In modern text-to-image systems, this process is often performed in latent space, conditioned on text embeddings, and followed by VAE decoding [6]. Although the mathematical details of diffusion training differ across frameworks, the inference bottleneck can be described in a simple form:

$$C_{infer} \approx N_{FE} \times C_{step} + C_{aux} \tag{1}$$

Here, C_{infer} denotes the overall cost of generating one image, N_{FE} denotes the number of denoising function evaluations, C_{step} denotes the cost of one denoising-network call, and C_{aux} denotes auxiliary overhead outside the core denoiser. This expression is not a strict hardware performance model. Rather, it is a conceptual tool for comparing methods. It clarifies that accelerating diffusion inference can be achieved by reducing the number of denoising steps, making each denoising step cheaper, reducing auxiliary overhead, or combining these directions.

The first term, N_{FE}, is the most visible source of cost. A standard sampler may require tens or hundreds of denoising evaluations. Fast samplers and schedule optimization methods aim to reduce this term while keeping the pretrained denoiser unchanged. The second term, C_{step}, depends on the model backbone, feature resolution, attention modules, numerical precision, and memory access. This term becomes especially important in high-resolution text-to-image generation, where one denoising call may involve large U-Net or Transformer-like networks. The third term, C_{aux}, includes text encoders, CFG branches, VAE decoding, scheduler logic, data movement, operator launch overhead, and deployment framework behavior. When N_{FE} becomes small, these auxiliary costs can occupy a larger share of end-to-end latency.

DOI: 10.54254/2755-2721/2026.LD36928

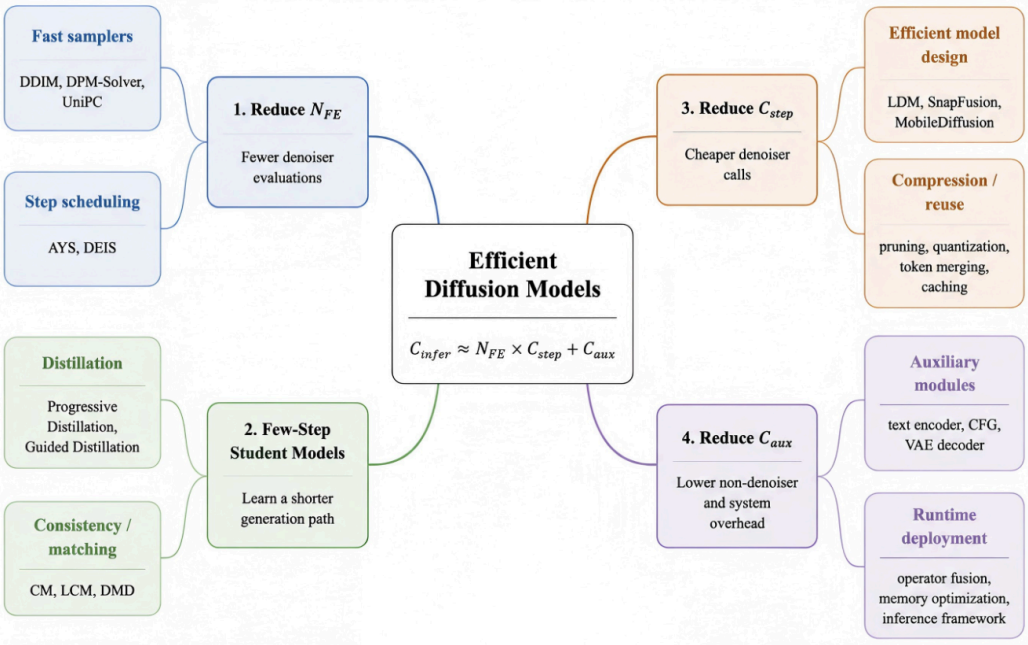

Figure 1. Cost-aware taxonomy of efficient diffusion models. The framework separates inference cost into N_{FE}, C_{step}, and C_{aux}, and uses these terms to organize representative acceleration routes

Figure 1 summarizes the proposed perspective. It shows that efficient diffusion methods do not all optimize the same object. Some methods mainly reduce sampling steps, some construct students that are trained for few-step generation, some reduce the computation within each step, and some target system-level overhead. This distinction matters because the same reported acceleration number may come from different sources and may not transfer across models, resolutions, or hardware platforms.

3. Efficient diffusion methods under the cost framework

3.1. Reducing sampling steps

Sampling-step reduction primarily targets N_{FE}. The basic idea is to obtain acceptable image quality with fewer denoising-network evaluations. DDIM is an early and influential example. By using a non-Markovian and optionally deterministic sampling process, it allows a pretrained diffusion model to sample along a sparse sequence of timesteps [4]. This reduces the number of denoising calls without changing the denoiser parameters. Its advantage is ease of integration, but the per-step cost remains unchanged and quality can degrade under extremely few steps.

The continuous-time view of diffusion models gives another route. Score-SDE formulates generation through stochastic differential equations and probability flow ordinary differential equations [5]. From this viewpoint, fast sampling becomes a numerical approximation problem. DPM-Solver designs high-order solvers for diffusion ODEs and can achieve good quality with fewer function evaluations [10]. EDM further emphasizes that noise parameterization, preconditioning, and sampling schedules strongly affect few-step quality [11]. These methods show that the location and numerical treatment of sampling steps can be as important as the raw number of steps.

The main advantage of sampler-based acceleration is that it is usually training-free. It can often be applied to existing image generation models with low integration cost. The limitation is also

DOI: 10.54254/2755-2721/2026.LD36928

clear: the denoising network is still the same network. If the base model is large, if CFG requires conditional and unconditional predictions, or if VAE decoding dominates latency, reducing N_{FE} alone may not be enough for real-time deployment.

3.2. Constructing few-step student models

Few-step student models also reduce N_{FE}, but they do so by changing the learned generation behavior rather than only changing the sampler. Progressive distillation trains a student model so that one student step approximates multiple teacher steps, gradually compressing the sampling chain [8]. This makes more aggressive step reduction possible than simple timestep skipping, but it introduces offline distillation cost and teacher dependence.

Consistency models provide another paradigm. They learn mappings that produce consistent predictions across noise levels, enabling one-step or few-step generation [12]. Latent consistency models adapt this idea to latent diffusion and are especially relevant for high-resolution text-to-image generation [13]. Guided diffusion distillation addresses the fact that many conditional generation systems rely on CFG, whose repeated conditional and unconditional passes increase inference cost [14]. More recent distillation routes, such as adversarial diffusion distillation and distribution matching distillation, further aim at high-quality generation in one to four steps [15, 16].

The strength of few-step student models is their low-latency potential. They are suitable for interactive preview, mobile generation, and applications where immediate response matters more than perfect fidelity. Their cost, however, is moved to training. They often require strong teachers, carefully designed objectives, large-scale data, and additional tuning. For rapidly changing text-to-image ecosystems, distillation may also lag behind new base models.

3.3. Reducing per-step computation

Reducing C_{step} is complementary to reducing N_{FE}. LDM is the most representative example. Instead of denoising directly in pixel space, it uses an autoencoder to map images into a lower-dimensional latent space and performs diffusion there [6]. Because repeated denoising happens on a compact latent grid, the cost of each denoising step is much lower than in full-resolution pixel space. This is one reason latent diffusion became a practical foundation for modern text-to-image systems.

Other methods reduce per-step cost through model design or compression. Mobile-oriented systems such as SnapFusion and MobileDiffusion combine efficient network structures, fewer steps, distillation, decoder optimization, and hardware-aware deployment to achieve fast text-to-image generation on constrained devices [17, 18]. Quantization reduces model storage and bandwidth and may accelerate inference when supported by hardware [19]. Feature reuse and caching methods, such as DeepCache, exploit redundancy across denoising steps and avoid recomputing stable features at every step [20].

These methods show that per-step acceleration is not equivalent to reducing parameter count. Latency also depends on attention computation, feature-map size, memory bandwidth, operator fusion, precision, and implementation details. A compressed model may not be faster if the target hardware cannot efficiently execute its operators. A caching method may be effective under one sampler but less reliable when timestep intervals become large. Therefore, C_{step} should be evaluated with wall-clock latency and memory usage, not only FLOPs or parameter counts.

DOI: 10.54254/2755-2721/2026.LD36928

3.4. Reducing auxiliary and system overhead

System-level optimization mainly targets C_{aux} and the real wall-clock form of C_{step}. In text-to-image systems, the denoiser is not the only module. Text encoding, CFG, scheduler execution, VAE decoding, memory allocation, data transfer, and runtime scheduling all contribute to latency. A simplified end-to-end view can be written as:

$$T_{e2e} \approx T_{text} + N_{FE} \cdot T_{denoise} + T_{decode} + T_{runtime} \tag{2}$$

This expression explains why two methods with the same N_{FE} can have different user-perceived speed. Framework-level optimizations such as graph compilation, operator fusion, mixed precision, memory planning, attention kernels, and hardware-specific runtimes can strongly affect deployment. On mobile and edge devices, energy consumption, memory limits, thermal behavior, and supported low-precision operators also influence whether an acceleration method is practical.

System optimization is sometimes treated as engineering detail, but it is part of the efficiency problem. When sampling is compressed to a few steps, the relative importance of text encoding, VAE decoding, and runtime overhead increases. Therefore, a complete evaluation of efficient diffusion methods should include end-to-end latency, peak memory, resolution, precision, batch size, hardware, and inference framework, not only algorithmic step count.

Table 1. Representative efficient diffusion routes under the cost-decomposition framework

Route	Representative methods	Main cost term	Training required	Main advantage	Main limitation
Fast sampling and schedule design	DDIM, DPM-Solver, EDM	N_{FE}	Usually no	Easy to apply to pretrained models	Per-step and auxiliary costs remain
Few-step student models	Progressive Distillation, Consistency Models, LCM, ADD, DMD	N_{FE} and guided inference cost	Usually yes	Very low-latency potential	Teacher dependence and training cost
Latent and lightweight modeling	LDM, SnapFusion, MobileDiffusion	C_{step}	Often yes	Reduces repeated denoising cost	Capacity, decoder, and hardware constraints
Compression and reuse	Quantization, DeepCache	C_{step}	Optional or calibration-based	Reduces memory, bandwidth, or redundant computation	Quality loss or implementation sensitivity
Deployment optimization	Runtime compilation, fused kernels, mixed precision	C_{aux} and real latency	No model retraining in many cases	Converts algorithmic gains into wall-clock gains	Hardware and framework dependence

DOI: 10.54254/2755-2721/2026.LD36928

4. Comparative discussion and open challenges

The cost-decomposition view suggests that efficient diffusion methods should not be ranked by a single metric. Reducing N_{FE} is valuable when the sampling chain is long, but its benefit decreases when each denoising step or auxiliary module dominates latency. Reducing C_{step} can improve high-resolution generation, but it may require architecture changes, compression, calibration, or hardware support. Reducing C_{aux} is essential for deployed systems, but its contribution may be invisible in papers that only report sampling steps or FLOPs.

A practical method choice depends on the target scenario. If the goal is to accelerate an existing pretrained model with minimal engineering cost, fast samplers are attractive. If the goal is interactive preview or mobile generation, few-step students and lightweight backbones become more important. If high-resolution quality is the priority, latent-space modeling and moderate sampler acceleration may be safer than extreme compression. If the goal is a production system, runtime optimization, memory planning, and hardware-specific kernels should be evaluated together with model-level methods.

Several challenges remain. First, fair evaluation is still difficult. Different papers report different combinations of NFE, FLOPs, parameters, latency, memory, image quality, and hardware settings. Future work should report end-to-end latency, peak memory, resolution, precision, batch size, sampler, CFG scale, hardware, and framework. Second, joint optimization is underdeveloped. Samplers, distillation, quantization, caching, and system optimization can interact in nontrivial ways. A method that works well for a 50-step sampler may not behave the same under a 4-step student. Third, few-step quality remains a bottleneck. One-step or few-step models are promising for preview and interactive use, but final high-fidelity generation may still require refinement or multi-stage pipelines. Finally, modern text-to-image backbones increasingly use Transformer or hybrid architectures. Efficient generation must therefore consider token length, attention cost, text encoders, decoders, and deployment runtimes together rather than only compressing the denoiser.

5. Conclusion

This paper presents a short survey of efficient diffusion models for image generation from a cost-decomposition perspective. The central view is that inference cost can be understood through N_{FE}, C_{step}, and C_{aux}. Fast samplers reduce the number of denoising evaluations. Few-step student models learn shorter generation paths but introduce training cost. Latent modeling, lightweight architectures, compression, and caching reduce the cost of each denoising call. System-level optimization reduces auxiliary overhead and determines whether theoretical acceleration becomes real wall-clock speed.

The main implication is that efficient diffusion is not simply faster sampling. A practical image generation system must allocate cost across algorithms, model structure, and deployment infrastructure. The proposed cost-decomposition view provides a compact framework for comparing efficient diffusion methods and for selecting acceleration strategies under different quality, latency, memory, and hardware constraints.

References

[1] Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., & Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning* (pp. 2256-2265).

[2] Song, Y., & Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*.

DOI: 10.54254/2755-2721/2026.LD36928

[3] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*.
[4] Song, J., Meng, C., & Ermon, S. (2021). Denoising diffusion implicit models. In *International Conference on Learning Representations*.
[5] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.
[6] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10684-10695).
[7] Ho, J., & Salimans, T. (2022). Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.
[8] Salimans, T., & Ho, J. (2022). Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*.
[9] Shen, H., Zhang, J., Xiong, B., Hu, R., Chen, S., Wan, Z., Wang, X., Zhang, Y., Gong, Z., Bao, G., Tao, C., Huang, Y., Yuan, Y., Zhang, M., & Zhang, M. (2025). Efficient diffusion models: A survey. *Transactions on Machine Learning Research*.
[10] 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. In *Advances in Neural Information Processing Systems*.
[11] Karras, T., Aittala, M., Aila, T., Laine, S. (2022). Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*.
[12] Song, Y., Dhariwal, P., Chen, M., Sutskever, I. (2023). Consistency models. In *International Conference on Machine Learning*.
[13] Luo, S., Tan, Y., Huang, L., Li, J., Zhao, H. (2023). Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*.
[14] Meng, C., Rombach, R., Gao, R., Kingma, D. P., Ermon, S., Ho, J., Salimans, T. (2023). On distillation of guided diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
[15] Sauer, A., Lorenz, D., Blattmann, A., Rombach, R. (2023). Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*.
[16] Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W. T., & Park, T. (2024). One-step diffusion with distribution matching distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6613-6623).
[17] Li, Y., Wang, H., Jin, Q., Hu, J., Chemerys, P., Fu, Y., Wang, Y., Tulyakov, S., & Ren, J. (2023). SnapFusion: Text-to-image diffusion model on mobile devices within two seconds. In *Advances in Neural Information Processing Systems*.
[18] Zhao, Y., Xu, Y., Xiao, Z., Jia, H., & Hou, T. (2024). MobileDiffusion: Instant text-to-image generation on mobile devices. In *European Conference on Computer Vision*.
[19] Li, X., Liu, Y., Lian, L., Yang, H., Dong, Z., Kang, D., Zhang, S., & Keutzer, K. (2023). Q-Diffusion: Quantizing diffusion models. In *IEEE/CVF International Conference on Computer Vision*.
[20] Ma, X., Fang, G., & Wang, X. (2024). DeepCache: Accelerating diffusion models for free. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 15762-15772).