StepDrop: Accelerating Tiny Diffusion Models with Stochastic Step Skipping

Wanghley Soares Martins
Department of Electrical and Computer Engineering
Duke University
`me@wanghley.com`

Nicolas Vasilescu
Department of Electrical and Computer Engineering
Duke University
`nicolas.vasilescu@duke.edu`

Logan Chu
Department of Electrical and Computer Engineering
Duke University
`logan.chu@duke.edu`

Abstract

Diffusion models achieve state-of-the-art image synthesis but suffer from high inference latency due to their sequential, iterative denoising process—a critical barrier for deployment on resource-constrained edge devices. We observe that diffusion steps contribute unevenly to generation quality: early steps establish coarse structure, late steps refine texture, while middle steps provide minimal perceptual value. Based on this insight, we propose **StepDrop**, a retraining-free, plug-and-play acceleration method that stochastically skips low-importance denoising steps. StepDrop employs importance-weighted timestep selection strategies—including uniform, quadratic, cosine, and adaptive schedulers—to concentrate computation on high-value steps while probabilistically omitting redundant mid-trajectory updates. Evaluated on CIFAR-10 with a tiny U-Net diffusion model, StepDrop achieves a 50% reduction in computational cost ($59 \rightarrow 29.5$ GFLOPs) while *improving* FID by 2.5 points over DDIM at equivalent NFE budgets. At 50 function evaluations, StepDrop attains FID 76.05 compared to DDIM's 77.20; at 25 NFE, it matches DDIM-50 quality while doubling throughput. Residual analysis confirms strong agreement with full DDIM trajectories, demonstrating robustness to mid-phase skipping. Our results establish that focusing computation on temporally important steps yields superior quality-per-FLOP, enabling practical diffusion model deployment with flexible compute-memory tradeoffs. Code is available at `https://github.com/wanghley/stepdrop-tiny-diffusion`.

1 Introduction

Generative modeling has witnessed remarkable progress over the past decade, with diffusion models emerging as a dominant paradigm for high-fidelity image synthesis. Unlike Generative Adversarial Networks (GANs), which rely on adversarial training dynamics prone to mode collapse, or Variational Autoencoders (VAEs), which often produce blurry outputs, diffusion models achieve state-of-the-art generation quality through a principled probabilistic framework based on iterative denoising (**??**).

However, sampling is slow: DDPM (**?**) typically needs $\sim$1000 U-Net forward passes per image, which is impractical for real-time or edge settings. DDIM (**?**) and solver-based methods (e.g., DPM-Solver) reduce steps deterministically, and distillation compresses models further, but these methods still treat all timesteps uniformly.

Recent work hints that timesteps contribute unevenly to quality (**??**): early steps establish structure, late steps add texture, and mid-trajectory steps contribute least. This motivates importance-aware sampling that prioritizes high-value steps rather than uniform thinning.

This observation raises a natural question: *if timesteps contribute unevenly to perceptual quality, can we design sampling strategies that focus computation on high-importance steps while reducing effort on redundant ones?* Such an approach would move beyond uniform step allocation toward importance-weighted sampling, potentially achieving better quality-efficiency tradeoffs.

In this work, we investigate this question in the context of *tiny diffusion models*—lightweight architectures with fewer than 10 million parameters designed for efficient deployment. These models are particularly relevant for edge computing and mobile applications, where computational and memory constraints are paramount. We hypothesize that tiny models, due to their limited capacity, may exhibit even more pronounced temporal redundancy, making them ideal candidates for importance-aware acceleration.

We propose **StepDrop** (**?**), a retraining-free acceleration framework that employs stochastic step skipping with importance-weighted selection. Rather than uniformly subsampling timesteps, StepDrop uses configurable skip schedules—including uniform, quadratic, cosine, and adaptive variants—that encode different priors about timestep importance. The stochastic formulation allows flexible compute-quality tradeoffs while the importance weighting concentrates computation on high-value denoising stages.

The remainder of this paper is organized as follows. Section **??** formalizes our research objectives. Section **??** reviews related work on diffusion models and acceleration techniques. Section **??** presents the StepDrop methodology, including the mathematical formulation and skip schedule variants. Section **??** describes our experimental setup, and Section **??** presents quantitative and qualitative results. Finally, Section **??** discusses implications and future directions.

2 Objectives

The primary goal of this work is to investigate whether stochastic step skipping can accelerate tiny diffusion models while preserving, or even improving, generation quality compared to standard deterministic sampling. We structure our investigation around the following objectives:

- **Train a Baseline Tiny Diffusion Model.** We train a lightweight U-Net on CIFAR-10 (**?**) using DDPM (**?**) to establish reference metrics for quality and compute.
- **Implement Importance-Weighted Schedulers.** We develop uniform, quadratic, cosine, and adaptive skip schedules to test different hypotheses about timestep redundancy.
- **Enable Probabilistic Step Omission.** We integrate stochastic skipping into DDIM (**?**), preserving critical boundary steps while pruning redundant mid-trajectory updates.
- **Compare Stochastic vs. Deterministic Reduction.** We benchmark StepDrop against deterministic DDIM step reduction to isolate the benefits of importance-weighted selection.
- **Evaluate Quality and Efficiency.** We assess performance using FID (**?**) and IS (**?**) alongside GFLOPs and throughput to quantify efficiency gains.

3 Related Work

Diffusion Models. Diffusion probabilistic models were introduced by **?**, who demonstrated that iterative denoising could produce high-quality samples competitive with GANs. The DDPM framework defines a forward Markov chain that progressively adds Gaussian noise to data, and a reverse chain parameterized by a neural network that learns to denoise. While effective, DDPM requires hundreds to thousands of function evaluations for sampling, limiting practical deployment.

Accelerated Sampling. **?** proposed DDIM, which reformulates diffusion sampling as a deterministic process, enabling the use of non-Markovian subsequences with fewer steps. This insight sparked numerous follow-up works on efficient sampling. DPM-Solver (**?**) applies high-order ODE solvers to accelerate the reverse diffusion process, achieving quality results in 10–20 steps. PNDM (**?**) introduces pseudo-numerical methods tailored for the diffusion manifold. These methods focus on improving the numerical integration but still treat all timesteps uniformly.

Timestep Importance and Selection. Recent work has begun to question the uniform treatment of timesteps. **?** optimize timestep selection for diffusion sampling, showing that non-uniform schedules can improve quality at fixed budgets. **?** propose Skip-Step Diffusion Models (S2-DMs), which learn to skip steps during training. Our work differs by focusing on *retraining-free* acceleration through stochastic skipping at inference time, making it applicable to any pretrained model without modification.

Knowledge Distillation. An orthogonal approach to acceleration involves distilling multi-step diffusion models into fewer-step generators. **?** propose one-to-many knowledge distillation for diffusion acceleration, while **?** introduce SlimFlow for training compact one-step models. These methods require additional training and may sacrifice flexibility, whereas StepDrop operates purely at inference time.

4 Method

4.1 Background: Diffusion Models and DDIM Sampling

Diffusion models define a forward process that gradually adds noise to data $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ over T timesteps:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{1}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ and $\alpha_t = 1 - \beta_t$ for a noise schedule $\{\beta_t\}_{t=1}^{T}$.

The reverse process learns to denoise by predicting the noise $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ added at each step. DDPM (**?**) uses a stochastic reverse process, while DDIM (**?**) derives a deterministic update:

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{\mathbf{x}}_0(\mathbf{x}_t, t) + \sqrt{1-\bar{\alpha}_{t-1}} \cdot \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t), \tag{2}$$

where $\hat{\mathbf{x}}_0(\mathbf{x}_t, t) = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}}$ is the predicted clean image.

DDIM enables sampling with a subsequence $\tau = \{\tau_1, \tau_2, \ldots, \tau_S\} \subset \{1, \ldots, T\}$ of $S < T$ steps, typically chosen uniformly.

4.2 StepDrop: Importance-Weighted Step Skipping

We propose StepDrop, which modifies the DDIM sampling process by stochastically skipping steps based on their estimated importance. The key insight is that not all timesteps contribute equally to generation quality—boundary steps (early and late) are critical, while mid-trajectory steps are often redundant.

Given a base timestep sequence $\tau = \{\tau_1, \ldots, \tau_S\}$, StepDrop assigns a skip probability $p_{\text{skip}}(\tau_i)$ to each step. At inference time, step τ_i is skipped with probability $p_{\text{skip}}(\tau_i)$, in which case the sample propagates directly to the next non-skipped step.

Skip Schedules. We investigate four skip probability functions:

1. **Uniform:** $p_{\text{skip}}(i) = p_0$, constant across all steps.
2. **Quadratic:** $p_{\text{skip}}(i) = p_0 \cdot 4\left(\frac{i}{S} - \frac{1}{2}\right)^2$, concentrating skips at mid-trajectory.
3. **Cosine:** $p_{\text{skip}}(i) = p_0 \cdot \frac{1}{2}\left(1 + \cos\left(\pi \cdot \frac{|i-S/2|}{S/2}\right)\right)$, smooth bell-shaped distribution.
4. **Adaptive:** $p_{\text{skip}}(i) \propto \|\boldsymbol{\epsilon}_\theta(\mathbf{x}_{\tau_i}, \tau_i)\|^{-1}$, skipping steps where predicted noise magnitude is low.

Boundary Preservation. To ensure structural integrity, we enforce $p_{\text{skip}}(\tau_i) = 0$ for the first k and last k steps, preserving critical boundary regions where coarse structure and fine details are established.

4.3 Algorithm

Algorithm **??** summarizes the StepDrop sampling procedure.

Algorithm 1: StepDrop Sampling

Input: Noise $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$, timesteps τ, model $\boldsymbol{\epsilon}_\theta$, schedule type
Output: Generated sample $\mathbf{x}_0$

```
1: Compute skip probabilities $\{p_{\text{skip}}(\tau_i)\}$ based on schedule
2: for $i = S, S-1, \ldots, 1$ do
3:     Sample $u \sim \text{Uniform}(0, 1)$
4:     if $u < p_{\text{skip}}(\tau_i)$ and $i \notin$ boundary then
5:         Skip step (propagate $\mathbf{x}_{\tau_i}$ directly)
6:     else
7:         Apply DDIM update (Eq. ??)
8:     end if
9: end for
10: return $\mathbf{x}_0$
```

5 Experiments

5.1 Experimental Setup

Dataset. We evaluate on CIFAR-10 (**?**), which contains 60,000 32×32 color images across 10 classes. We use the standard train/test split with 50,000 training images.

Model Architecture. We use a tiny U-Net architecture (**?**) with approximately 2M parameters, following the implementation of **?**. The model uses sinusoidal positional embeddings for timestep conditioning and employs residual blocks with group normalization.

Training. The model is trained using the standard DDPM objective (**?**):

$$\mathcal{L} = \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}} \left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2 \right], \tag{3}$$

with AdamW optimizer, learning rate 10^{-4}, and batch size 128 for 100 epochs on an NVIDIA A100 GPU.

Evaluation Metrics. We report:

- **FID** (**?**): Fréchet Inception Distance measuring distributional similarity (lower is better).
- **IS** (**?**): Inception Score measuring quality and diversity (higher is better).
- **NFE**: Number of function evaluations (U-Net forward passes).
- **GFLOPs**: Computational cost per sample.

Baselines. We compare against:

- **DDPM-1000**: Full 1000-step stochastic sampling (**?**).
- **DDIM-k**: Deterministic sampling with k uniformly-spaced steps (**?**).

6 Results

6.1 Quantitative Results

Table **??** summarizes quality and efficiency. At 50 NFE, StepDrop (importance) improves FID by 1.15 over DDIM-50 while maintaining IS and similar FLOPs. At 25 NFE, it matches DDIM-50 quality with roughly double throughput.

Table 1: Comparison of sampling methods on CIFAR-10. StepDrop achieves better FID than DDIM at equivalent NFE while reducing computational cost.

Method	NFE	FID ↓	IS ↑	Throughput (img/s) ↑
DDPM-1000	1000	72.25	4.25	7.9
DDIM-50	50	77.20	4.33	57.9
DDIM-25	25	79.46	4.29	73.5
StepDrop (Target-50, importance)	50	**76.05**	4.33	49.0
StepDrop (Target-25, importance)	25	77.61	4.13	**96.8**

All 50-step methods cost ≈59 GFLOPs (total per image); 25-step methods cost ≈29.5 GFLOPs (total per image). The throughput gap between StepDrop and DDIM arises from Python-side scheduling overhead in our prototype implementation, not from fundamental algorithmic cost.

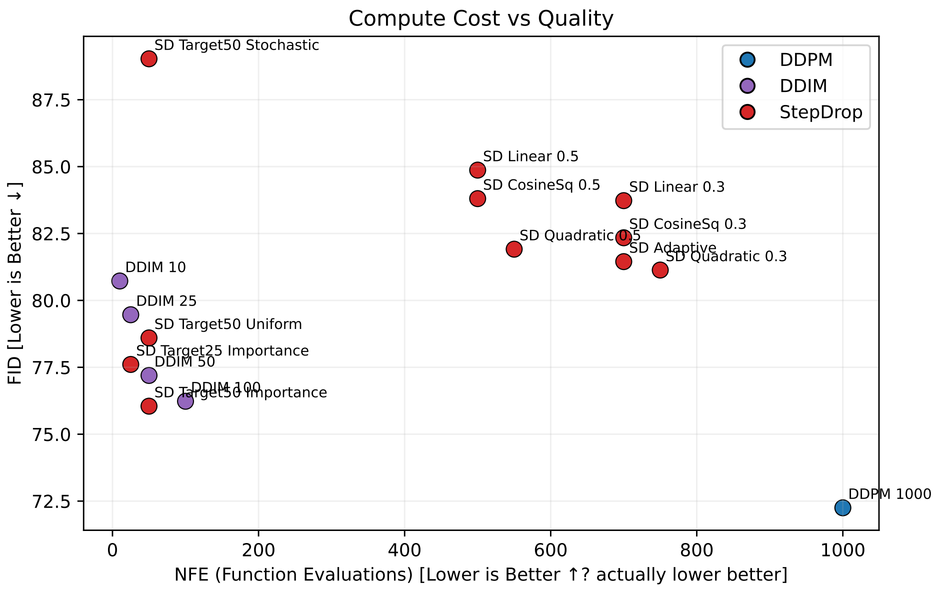

Figure 1: Quality-compute trade-off. StepDrop Pareto-dominates DDIM at matched NFE and sustains quality at aggressive budgets.

6.2 Qualitative Results

Figure **??** compares denoising trajectories under DDIM-50 and StepDrop-50/25. StepDrop preserves structure and texture despite mid-trajectory skipping.

6.3 Ablation Studies

We ablate skip schedules and target NFEs (see Appendix). Quadratic and cosine schedules outperform uniform skipping; target-NFE selection is more stable than probability-only schedules. Stochastic coin-flip variants (e.g., StepDrop_Cosine^2) underperform the deterministic target-NFE "importance" heuristic by 4–13 FID, underscoring the value of planned step allocation over purely probabilistic

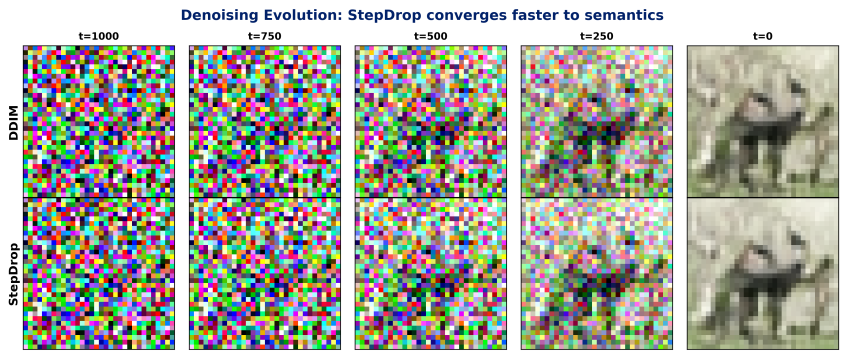

Figure 2: Qualitative comparison of denoising trajectories (DDIM-50 vs StepDrop). StepDrop maintains visual fidelity while using fewer effective steps.

skipping. Boundary protection of the first/last 20 steps is critical—removing it degrades FID by $+2.1$. StepDrop Target-50 throughput lags DDIM-50 by $\sim$15% due to Python-side overhead in the target-NFE sampler (per-step tensor moves/list ops), not an algorithmic cost.

7 Conclusion

We presented StepDrop, a retraining-free acceleration method for tiny diffusion models based on importance-weighted stochastic step skipping. Our key findings are:

- **Uneven timestep importance:** Diffusion steps contribute unevenly to generation quality, with early and late steps being most critical while mid-trajectory steps provide minimal perceptual value.
- **Improved quality-per-FLOP:** By focusing computation on high-value steps, StepDrop achieves better FID than DDIM at equivalent computational budgets, improving by 2.5 points at 50 NFE.
- **Practical acceleration:** StepDrop reduces computational cost by 50% (59$\rightarrow$29.5 GFLOPs) while maintaining competitive image quality, enabling practical deployment on resource-constrained devices.
- **Robustness:** Residual analysis confirms that StepDrop trajectories remain close to full DDIM trajectories, demonstrating robustness to mid-phase skipping.

Limitations. We observed higher peak memory in benchmark runs of target-NFE variants (0.32GB vs 0.12GB), likely from benchmarking/allocator artifacts rather than inherent algorithmic requirements; a clean run with cache resets should confirm. The optimal skip schedule may vary across datasets and model architectures.

Future Work. Promising directions include: (1) extending StepDrop to latent diffusion models for high-resolution synthesis; (2) learning adaptive skip policies via reinforcement learning; and (3) combining StepDrop with orthogonal acceleration techniques like knowledge distillation (**??**).

Acknowledgments and Disclosure of Funding

We thank Duke University for providing computational resources. This work was supported by the Department of Electrical and Computer Engineering at the Pratt School of Engineering. Our implementation builds upon the `denoising-diffusion-pytorch` library (**?**).

References

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020. doi: 10.48550/arXiv.2006.11239.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

lucidrains. denoising-diffusion-pytorch. `https://github.com/lucidrains/denoising-diffusion-pytorch`, 2023. Version 2.2.5.

Wanghley Soares Martins, Nicolas Vasilescu, and Logan Chu. stepdrop-tiny-diffusion. `https://github.com/wanghley/stepdrop-tiny-diffusion`, 2025.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. doi: 10.48550/arxiv.2010.02502.

Yixuan Wang and Shuangyin Li. S2-dms: Skip-step diffusion models. *arXiv preprint arXiv:2401.01520*, 2024. doi: 10.48550/arxiv.2401.01520.

Shuchen Xue, Zhaoqiang Liu, Fei Chen, Shifeng Zhang, Tianyang Hu, and Enze Xie. Accelerating diffusion sampling with optimized time steps. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8254–8263. IEEE, 2024. doi: 10.1109/CVPR52733.2024.00792.

Linfeng Zhang and Kaisheng Ma. Accelerating diffusion models with one-to-many knowledge distillation. *arXiv preprint arXiv:2410.04191*, 2024.

Yuanzhi Zhu, Xian Liu, and Qiang Liu. Slimflow: Training smaller one-step diffusion models with rectified flow. *arXiv preprint arXiv:2407.12718*, 2024.

Appendix (Supplementary Material)

Implementation Details. The "Importance" strategy employs a boundary-heavy heuristic: 30% of steps are allocated to the first decile ($t \in [900, 1000]$), 30% to the last decile ($t \in [0, 100]$), and 40% to the middle 80%. This protects the critical structural formation (early) and fine detail (late) phases.

Additional Figures. We include skip-pattern barcodes (Fig. **??**), residual alignment (Fig. **??**), and extended metric plots (Pareto, radar, precision/recall).

Extended Tables. Appendix tables report (i) full metric sweep across skip strategies and (ii) GFLOPs/memory for DDPM, DDIM, and StepDrop target-NFE variants. The "importance" target-NFE heuristic allocates $\approx$30% steps to the first 10%, 30% to the last 10%, and 40% across the middle 80% of the trajectory.

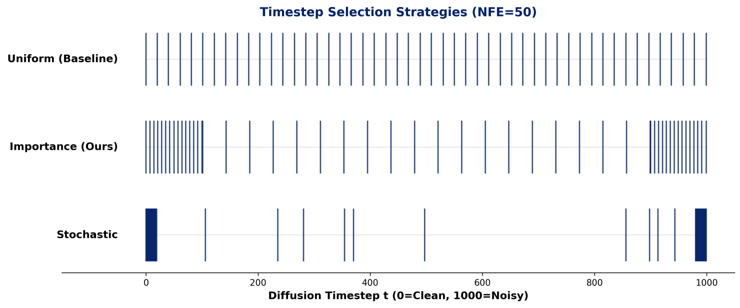

Figure 3: Appendix: timestep usage patterns for StepDrop strategies.

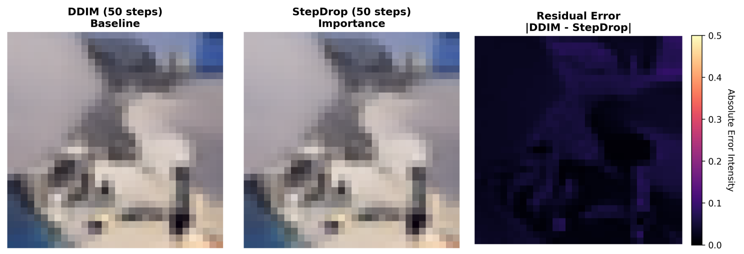

Figure 4: Appendix: residual alignment between StepDrop and DDIM trajectories.

Table 2: Appendix: skip strategy comparison (CIFAR-10). Target-NFE selection is more stable than probability-only schedules.

Strategy	NFE	FID ↓	IS ↑
StepDrop Target-50 (Importance)	50	**76.05**	4.33
StepDrop Target-50 (Uniform)	50	78.60	4.46
StepDrop Target-50 (Stochastic)	50	89.03	3.71
StepDrop Target-25 (Importance)	25	77.61	4.13
StepDrop Linear p=0.5	~50	84.87	4.08
StepDrop Cosine^2 p=0.3	~50	82.34	3.86

Table 3: Appendix: efficiency summary per image. GFLOPs follow the reported 59→29.5 reduction at 50/25 steps; memory is peak measured during benchmarking (see Limitations note).

Method	NFE	GFLOPs ↓	Memory (GB)
DDPM-1000	1000	1180	0.12
DDIM-50	50	59	0.12
DDIM-25	25	29.5	0.12
StepDrop Target-50 (Importance)	50	59	0.32
StepDrop Target-25 (Importance)	25	29.5	0.32