StepDrop: Accelerating Tiny Diffusion Models with Stochastic Step Skipping

Wanghley Soares Martins[1], Nicolas Vasilescu[1], Logan Chu[1]
[1]Department of Electrical and Computer Engineering, Pratt School of Engineering

Introduction

High inference latency is the principal barrier to the widespread deployment of powerful tiny diffusion models. While these models exhibit impressive performance in image synthesis, their sequential, iterative denoising process incurs substantial computational cost, which is prohibitive for resource-constrained edge devices and real-time applications. This work is motivated by the critical need for efficient sampling techniques that can drastically accelerate inference without a commensurate degradation in the fidelity of the generated output. Developing a solution that maintains high-quality generation of DDPM/DDIM while ensuring practicality and speed is essential for advancing the applicability of these models.

Objectives

- Train a baseline tiny diffusion model (DDPM/DDIM).
- Implement stochastic and adaptive step-skip schedulers.
- Enable probabilistic omission of denoising steps.
- Compare stochastic vs. deterministic step reduction.
- Evaluate using standard image-quality metrics.

Method

Training:

Dataset: CIFAR-10 | Loss: MSE on noise prediction | Optimizer: AdamW

Step-Skipping Schedulers

1. **Uniform**: skips occur evenly across all timesteps.
2. **Quadratic**: skips are strongly concentrated in the middle.
3. **Cosine**: skips are smooth and most frequent toward the middle.
4. **Adaptive**: dynamically selects steps based on predicted noise magnitude

Stochastic Step Dropping (based on Wang et al., 2024)

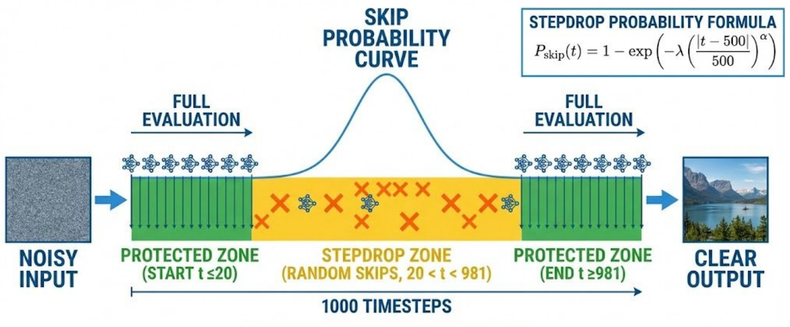

Figure 1: Boundary preservation and stochastic mid-step skipping flow.
Source: by the authors, 2025

DDPM (Ho et al., 2020): Stochastic, Markovian sampler

DDIM (Song et al., 2020): Deterministic, Non-Markovian sampler

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_\theta(x_t, t)\right) + \sigma_t z \qquad x_{t'} = \sqrt{\bar{\alpha}_{t'}}\,\hat{x}_0 + \sqrt{1-\bar{\alpha}_{t'}}\,\epsilon_\theta(x_t, t)$$

Evaluation Metrics

- **FID** (Fréchet Inception Distance): Measures the distance between feature distributions of generated and real images.
- **IS** (Inception Score): Measures both **quality** (recognizability) and **diversity** (content variety).

Results

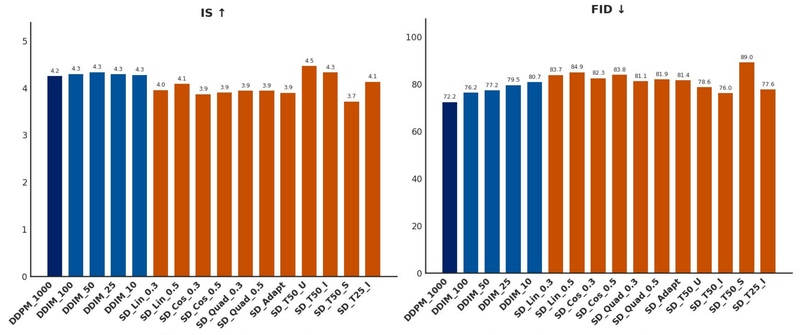

Figure 2: Image Quality Metrics Across Sampling Methodsflow.
Source: by the authors, 2025

- **Quality under compression:** StepDrop consistently outperforms uniform skipping at the same NFE.
- **50 NFE**: Achieves FID 76.05, improving over DDIM's 77.20 with identical compute.
- **25 NFE**: Matches DDIM-50 quality while doubling throughput.
- **Compute efficiency**: Skipping UNet calls reduces cost from 59 → 29.5 GFLOPs, yielding proportional energy savings for edge deployment.

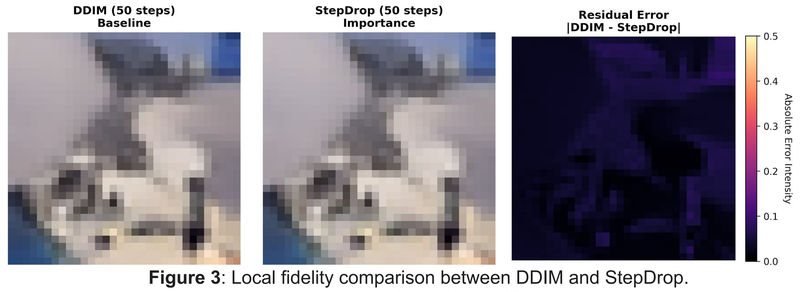

Figure 3: Local fidelity comparison between DDIM and StepDrop.
Source: by the authors, 2025

- **Diffusion steps vary in importance**, with clear temporal structure in their contribution to image quality.
- **Importance-based StepDrop improves FID by 2.5**, showing early and late steps matter most, while mid-trajectory steps add little.
- The model is **robust to skipping during the mid "drift" phase**, where updates have minimal perceptual impact.
- Focusing **computation on high-value steps** yields better image quality per compute and outperforms rigid step schedules.

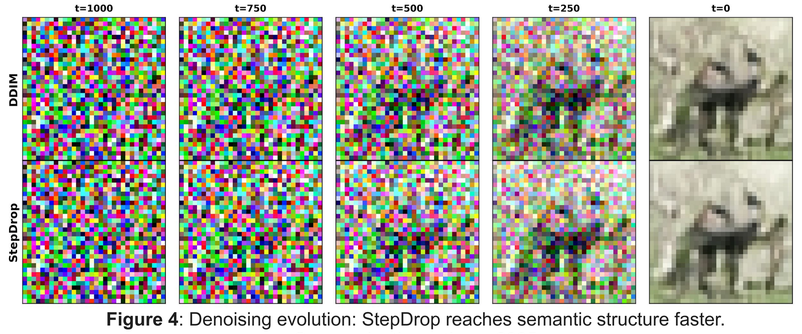

Figure 4: Denoising evolution: StepDrop reaches semantic structure faster.
Source: by the authors, 2025

Conclusions

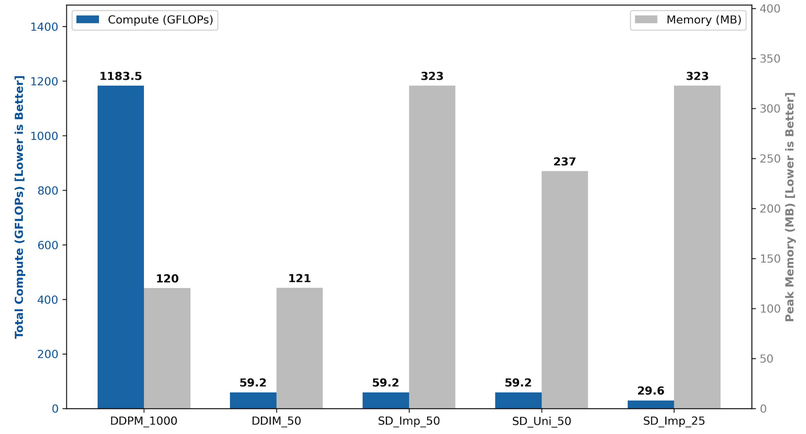

Figure 5: StepDrop efficiency–performance tradeoff.
Source: by the authors, 2025

- **Diffusion cost stems from uniform sampling**, despite uneven step importance (early = structure, late = texture, middle = low-value).
- **StepDrop targets high-value steps**, cutting compute by 50% while improving FID (+2.5) and keeping IS stable.
- Offers **best fidelity per FLOP** with flexible compute–memory tradeoffs; residuals show strong agreement with DDIM.
- **Practical**: zero-training, plug-and-play acceleration for any diffusion model; boosts edge viability and reduces serving energy.
- **Future**: extend to latent diffusion, build adaptive kernels, and reduce the ~3× memory overhead from precomputed timestep schedules

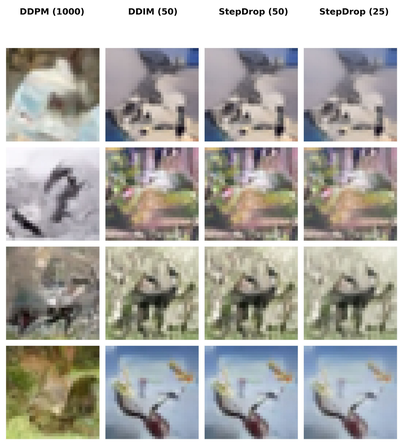

Figure 6:Strategies generation comparison.
Source: by the authors, 2025

References

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *arXiv*. https://doi.org/10.48550/arXiv.2006.11239

lucidrains. (2023). *denoising-diffusion-pytorch* (Version 2.2.5) [Computer software]. GitHub. https://github.com/lucidrains/denoising-diffusion-pytorch

Martins, W.S., Vasilescu, N., & Chu, L. (2025). stepdrop-tiny-diffusion [Computer software]. GitHub. https://github.com/wanghley/stepdrop-tiny-diffusion

Song, J., Meng, C., & Ermon, S. (2020). Denoising diffusion implicit models. *arXiv*. https://doi.org/10.48550/arxiv.2010.02502

Wang, Y., & Li, S. (2024). S2-DMs: Skip-step diffusion models. *arXiv*. https://doi.org/10.48550/arxiv.2401.01520

Xue, S., Liu, Z., Chen, F., Zhang, S., Hu, T., & Xie, E. (2024). Accelerating diffusion sampling with optimized time steps. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 8254–8263). IEEE. https://doi.org/10.1109/CVPR52733.2024.00792