Early Exiting for Accelerated Inference in Diffusion Models

Taehong Moon [1] **Moonseok Choi** [1] **EungGu Yun** [2] **Jongmin Yoon** [1] **Gayoung Lee** [3] **Juho Lee** [1]

Abstract

Diffusion models have achieved impressive results in generating content across domains like images, videos, text, and audio. However, their sampling speed is a practical challenge due to repeated evaluation of score estimation networks during inference. To address this, we propose a novel framework that optimizes compute allocation for score estimation, reducing overall sampling time. Our key insight is that the computation required for score estimation varies at different time steps. Based on this observation, we introduce an early-exiting scheme that selectively skips the subset of parameters in the score estimation network during the inference, guided by a time-dependent exit schedule. We apply this technique to image synthesis with diffusion models and demonstrate significantly improved sampling throughput without compromising image quality. Moreover, our approach seamlessly integrates with various types of solvers for faster sampling, leveraging their compatibility to enhance overall efficiency.

1. Introduction

Diffusion models have emerged as standard generative models due to their powerful performance across diverse domains including image synthesis (Ho et al., 2020; Dhariwal & Nichol, 2021; Ho et al., 2022a), 3D point cloud generation (Luo & Hu, 2021), text-to-image generation (Ramesh et al., 2022; Rombach et al., 2022), text-to-speech generation (Jeong et al., 2021), and video generation (Ho et al., 2022b). These models learn the reverse process of injecting noise into the data and denoise inputs progressively during sampling based on the learned model.

[1]Kim Jaechul Graduate School of AI, Korea Advanced Institute of Science and Technology, Seoul, South Korea [2]Saige Research, Seoul, South Korea [3]Naver AI Lab, Seongnam, Gyeonggi-do, South Korea. Correspondence to: Juho Lee <juholee@kaist.ac.kr>.

Accepted to ICML workshop on Structured Probabilistic Inference & Generative Modeling., Honolulu, Hawaii, USA. PMLR 202, 2023.

A major drawback of diffusion models is their slow sampling speed, as they require multiple steps of forward passes to generate a single sample, unlike the other methods such as GANs (Goodfellow et al., 2014) that can generate samples with only a single forward pass. To tackle this challenge, various approaches, such as enhancing ODE/SDE solvers (Kong & Ping, 2021; Lu et al., 2022; Zhang & Chen, 2023) and distilling models that require fewer sampling steps (Salimans & Ho, 2022; Song et al., 2023), have been proposed to minimize the number of steps needed for sampling diffusion models. Additionally, in line with the current trend of scaling large models across different domains, diffusion models with a substantial number of parameters are gaining popularity due to their ability to generate high-quality samples (Peebles & Xie, 2022). Running such large diffusion models for a number of sampling steps results in substantial computational overhead, underscoring the need for additional research to optimize computations and allocate resources efficiently.

In this paper, we introduce Adaptive Score Estimation (ASE) for faster sampling from diffusion models, inspired by the successful utilization of early-exiting schemes in Large Language Models (LLMs) (Schuster et al., 2022; Hou et al., 2020; Liu et al., 2021; Schuster et al., 2021). We hypothesize that the difficulty of score estimation may vary at different time steps, and based on this insight, we adapt the computation of blocks differently for each time step. As a result, this approach allows us the capability to dynamically control computation time during the sampling process. To achieve this, we introduce a time-varying block-dropping schedule and a simple algorithm for fine-tuning diffusion models to optimize them accordingly. ASE successfully accelerates the sampling speed while preserving high-quality samples. Furthermore, ASE is highly versatile, as it can be applied to various backbone architectures and can be combined with different solvers to further enhance sampling speed. We validate the effectiveness of our method through experiments on real-world image synthesis tasks.

2. Methods

This section describes our main contribution - Adaptive Score Estimation (ASE) for diffusion models. The section is organized as follows. We first briefly give a recap on

how to train a diffusion model and provide our intuition on the time-varying complexity of score estimation. Then we propose several options for time-dependent exit schemes of ASE, and present our main algorithm to accelerate the inference of the diffusion model.

2.1. Time-Varying Complexity of Score Estimation

Training diffusion models Let $x_0 \sim p_{\text{data}}(x) := q(x)$ be a sample from a target data distribution. In a diffusion model, we build a Markov chain that gradually injects Gaussian noises to x_0 to turn it into a sample from a noise distribution $p(x_T)$, usually chosen as standard Gaussian distribution. Specifically, given a noise schedule $(\beta_t)_{t=1}^T$, the forward process of a diffusion model is defined as

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I), \tag{1}$$

$$q(x_T|x_0) \approx \mathcal{N}(x_T; 0, I). \tag{2}$$

Then we define a backward diffusion process with a parameter θ as,

$$p_\theta(x_{1:T}) := p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \,|\, x_t), \tag{3}$$

so that we can start from $x_T \sim \mathcal{N}(0, I)$ and denoise it into a sample x_0. The parameter θ can be optimized by minimizing the negative of the lower-bound on the log-evidence,

$$\begin{aligned}\mathcal{L}_\theta &:= -\sum_{t=1}^{T} \mathbb{E}_q \left[D_{\text{KL}}[q(x_{t-1} \,|\, x_t, x_0) \| p_\theta(x_{t-1} \,|\, x_t)] \right] \\ &\geq -\log p_\theta(x_0)\end{aligned} \tag{4}$$

where

$$q(x_{t-1}|x_t, x_0) := \mathcal{N}\left(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I\right), \tag{5}$$

$$\tilde{\mu}_t(x_t, x_0) := \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \varepsilon_t \right). \tag{6}$$

When the model distribution $p_\theta(x_{t-1} \,|\, x_t)$ is chosen as a Gaussian, then above loss function can simplify as follows:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \varepsilon_\theta(x_t, t) \right) \tag{7}$$

$$\mathcal{L}(\theta) := \sum_{t=1}^{T} \mathbb{E}_{x_0, \varepsilon_t} \left[\lambda(t) \left\| \varepsilon_t - \varepsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon_t, t) \right\|^2 \right] \tag{8}$$

where $\lambda(t) = \frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1-\bar{\alpha}_t)}$. The neural network $\varepsilon_\theta(x_t, t)$ takes a corrupted sample x_t and estimates the noise that might have applied to a clean sample x_0.

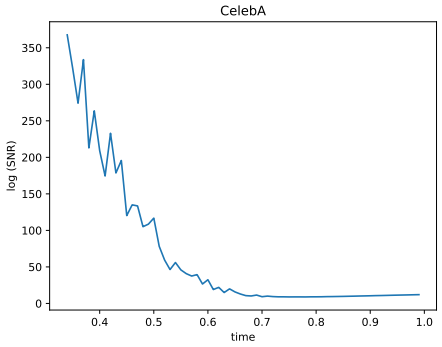

Figure 1. The $\log SNR$ of the score function $\nabla_{x_t} \log q(x_t)$ which is calculated with Monte-Carlo estimation of $\mathbb{E}_{x_0 \,|\, x_t} [\nabla_{x_t} \log q(x_t \,|\, x_0)]$ and $\text{Var}_{x_0 \,|\, x_t} [\nabla_{x_t} \log q(x_t \,|\, x_0)]$ computed with regard to CelebA.

To regard the training of diffusion models as a regression problem, we simplify the loss function through reparameterization

$$\mathcal{L}(\theta) = \sum_{t=1}^{T} \mathbb{E}_{x_0, x_t} \left[\lambda_t' \| \nabla_{x_t} \log q(x_t \,|\, x_0) - s_\theta(x_t, t) \|^2 \right], \tag{9}$$

$$\nabla_{x_t} \log q(x_t \,|\, x_0) = -\frac{\varepsilon_t}{\sqrt{1-\bar{\alpha}_t}} \approx -\frac{\varepsilon_\theta(x_t, t)}{\sqrt{1-\bar{\alpha}_t}} := s_\theta(x_t, t) \tag{10}$$

where $s_\theta(x_t, t)$ is the score estimation network. The optimal regressor of the score function $\nabla_{x_t} \log q(x_t)$ at time step t is obtained by taking the expectation of the conditional score function over the noiseless distribution $\mathbb{E}_{x_0 \,|\, x_t} [\nabla_{x_t} \log q(x_t \,|\, x_0)] = \nabla_{x_t} \log q(x_t)$.

To demonstrate the time-varying difficulty of score estimation, we measure the signal-to-ratio (SNR) plot of the score function. As depicted in the Figure 1, it unveils a clear pattern: high-SNR near the data distribution and low-SNR near the prior distribution. This indicates that more precise learning of the score function is required in the vicinity of the data distribution while it is relatively easy task to learn the score of data distribution near prior.

Adaptive Computation for Score Estimation To get the samples from diffusion models, we can apply Langevin dynamics to get samples from the distribution given the score function $\nabla_x \log p(x)$. Depending on the number of iteration N and step size β, we can iteratively update x_t as follows:

$$x_{t+1} = x_t + \beta \nabla_x \log p(x_t) + \sqrt{2\beta} z_t, \tag{11}$$

where $z_t \sim \mathcal{N}(0, I)$.

Due to this iterative evaluation, the total sampling time can be roughly be computed as $T \times \tau$, where T is the number

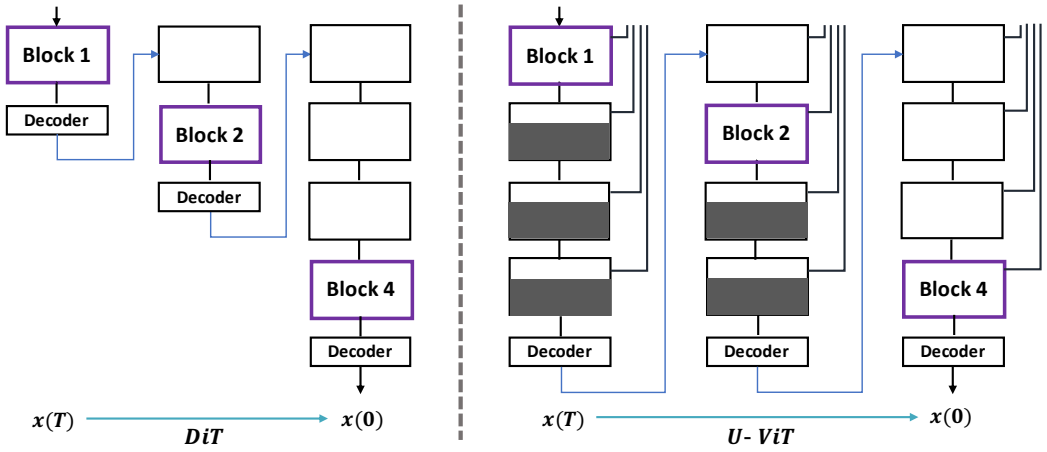

Figure 2. Schematic for the dropping schedules of DiT (left) and U-ViT (right). Due to the existence of residual connections in U-ViT, dropping encoder or decoder blocks in a straightforward manner cause severe performance degradation. In the case of U-ViT, the decoder blocks, except for the linear layer connected to encoder residual connections, are dropped.

of sampling steps and τ is the processing of diffusion model per time step. To enhance sampling efficiency, conventional approaches aim to reduce the number of time steps within the constrained value of τ.

In contrast, our method focus on reducing the processing time τ while maintaining accurate predictions within a given time interval. To accomplish this, we introduce adaptive score estimation, wherein the diffusion model dynamically allocates parameters based on the time t. For challenging task such as time $t \to 0$, the full parameter is utilized, while it induces skipping the subset of parameters for easier tasks. Additionally, we provide empirical evidence to support our idea in the Appendix A.1.

2.2. Adaptive Score Estimation for Diffusion Models

In the context of early exiting scheme in language models, CALM (Schuster et al., 2022) dynamically determines the exiting block depending on the state of intermediate features during the sampling process. On the other hand, our method pre-determines the exiting block based on the timestep t before sampling procedure. In particular, we have designed schedules for dropping the subset of parameters as time $t \to 1$ while preserving the parameters as time $t \to 0$. To reduce sampling time, it is preferable to skip as many blocks as possible. However, since we cannot determine the exact amount of parameters required for score estimation at each time t, we aimed to study diverse schedules that vary the amount of parameters dropped depending on time t. More detailed schedules that applied to both DiT and U-ViT are provided in Appendix A.2.

Following the removal of blocks based on the pre-defined schedule, we further fine-tune the model for a small number of iterations. This is attributed to the early exit approach, where the intermediate outputs of each building block are directly connected to the decoder. Consequently, the decoder receives input values that deviate from the distribution it observed during its initial training, necessitating adaptations. In the Appendix A.3, we provide the detailed fine-tuning algorithm.

During the sampling procedure, we induce early-exiting for the iterative evaluation of the diffusion model, as depicted in the Figure 2. More specifically, we apply ASE into both DiT and U-ViT architecture, considering the characteristic of each architecture. For the DiT model trained on ImageNet, consisting of 28 blocks, we designed a dropping schedule starting from the final block. In U-ViT, the dropping schedule had two main distinctions from DiT: the selection of candidate modules to drop and the subset of parameters to be skipped. Unlike DiT, we limited dropping to the decoder part in U-ViT. This decision was motivated by the presence of symmetric long skip connections between encoder and decoder, as dropping encoder modules induce the substantial information loss. Moreover, when dropping the parameters in U-ViT, we preserved the nn.Linear of a building block to retain feature information connected through skip connections, while skipping the remaining parameters.

3. Experiments

To verify the effectiveness of our method, we introduce a total of six early-exiting schedules as part of our experimental setup to validate the main hypothesis of our paper: the estimation of score function near the noise is relatively easy. In order to highlight the versatility of our method, we provide in-depth experiments where fast sampling techniques are

Table 1. DiT results using DDPM sampler and U-ViT results using EM sampler

Methods	DDPM-250	
	FID (↓)	Accel. (↑)
DiT	9.078	-
D1-DiT	8.747	16.63%
D3-DiT	8.662	23.43%
D4-DiT	**8.647**	30.46%
D6-DiT	9.764	38.32%

Methods	EM-1000	
	FID (↓)	Accel. (↑)
U-ViT	4.858	-
D1-U-ViT	**4.738**	16.26%
D3-U-ViT	4.819	24.77%
D4-U-ViT	4.851	28.03%
D6-U-ViT	5.044	32.09%

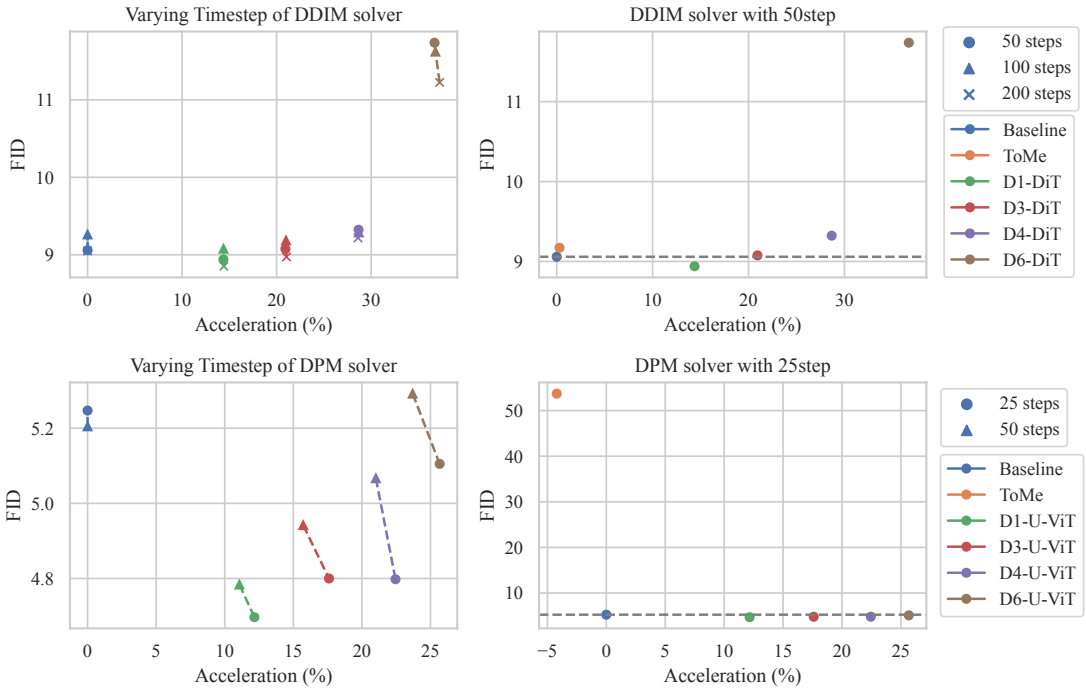

Figure 3. Performance with reduced timesteps when our method is combined with the fast sampling solver is shown in the first column. The second column of the figure illustrates the comparison between our method and the baseline. Each row corresponds to the results obtained on ImageNet and CelebA, respectively.

integrated with our fine-tuned DiT (Peebles & Xie, 2022) on ImageNet. Furthermore, we investigate whether our method is applicable to a different class of architecture, U-ViT (Bao et al., 2022). In the Appendix B, we provide the detailed setting of following experiments.

In Table 1, we observe that our approach effectively achieves faster inference while utilizing fewer parameters, yet maintains the same level of performance. Moreover, in Figure 3, we demonstrate that our method can be successfully combined with other fast sampling approaches, such as DDIM (Song et al., 2020) and DPM solver (Lu et al., 2022) with varying timesteps.

4. Conclusion and Limitations

In this paper, we present a novel method that effectively reduces the wallclock time by using an early-exiting scheme in diffusion models. Specifically, our method adaptively selects the blocks involved in denoising the inputs at each time step, taking into account the assumption that fewer parameters are required for early denoising steps. Surprisingly, we demonstrate that our method maintains performance in terms of FID scores even when reducing calculation costs by 30%. Our approach is not limited to specific architectures, as we validate its effectiveness on both U-ViT and DiT models. One limitation of our proposed method is that we manually design the schedule for the early-exiting scheme. As future work, we acknowledge the need to explore automated methods for finding an optimal schedule.

References

Bao, F., Li, C., Cao, Y., and Zhu, J. All are worth words: a vit backbone for score-based diffusion models. *arXiv preprint arXiv:2209.12152*, 2022.

Bolya, D. and Hoffman, J. Token merging for fast stable diffusion. *arXiv preprint arXiv:2303.17604*, 2023.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. pp. 1877–1901, 2020.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (ACL)*, 2019.

Dhariwal, P. and Nichol, A. Diffusion models beat GANs on image synthesis. pp. 8780–8794, 2021.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. 2021.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. 2014.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. pp. 6840–6851, 2020.

Ho, J., Saharia, C., Chan, W., Fleet, D. J., Norouzi, M., and Salimans, T. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022a.

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. *arXiv:2204.03458*, 2022b.

Hou, L., Huang, Z., Shang, L., Jiang, X., Chen, X., and Liu, Q. Dynabert: Dynamic bert with adaptive width and depth. 2020.

Jeong, M., Kim, H., Cheon, S. J., Choi, B. J., and Kim, N. S. Diff-tts: A denoising diffusion model for text-to-speech. In *International Speech Communication Association*, 2021.

Kong, Z. and Ping, W. On fast sampling of diffusion probabilistic models. In *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models (INNF+ 2021)*, 2021.

Liu, Y., Meng, F., Zhou, J., Chen, Y., and Xu, J. Faster depth-adaptive transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 13424–13432, 2021.

Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., and Hu, H. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3202–3211, 2022.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. 2022.

Luo, S. and Hu, W. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. pp. 8162–8171, 2021.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. 2022.

Schuster, T., Fisch, A., Jaakkola, T., and Barzilay, R. Consistent accelerated inference via confident adaptive transformers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Schuster, T., Fisch, A., Gupta, J., Dehghani, M., Bahri, D., Tran, V., Tay, Y., and Metzler, D. Confident adaptive language modeling. 2022.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. pp. 2256–2265, 2015.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. 2019.

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. Consistency models. 2023.

Strudel, R., Garcia, R., Laptev, I., and Schmid, C. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7262–7272, 2021.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. 2017.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. Segformer: Simple and efficient design for semantic segmentation with transformers. pp. 12077–12090, 2021.

Yang, X., Shih, S.-M., Fu, Y., Zhao, X., and Ji, S. Your ViT is secretly a hybrid discriminative-generative diffusion model. *arXiv preprint arXiv:2208.07791*, 2022.

Zhang, Q. and Chen, Y. Fast sampling of diffusion models with exponential integrator. 2023.

A. Adaptive Score Estimation

A.1. Empirical Validation of Our Hypothesis

To validate the hypothesis presented in Section 2.1, we devise two schedules for accelerating the inference of diffusion models: Noisy-Easy schedule, which is aligned with intuition, and the Data-Easy schedule, which assumes that score estimation near data is easier. The Noisy-Easy schedule allocates fewer parameters for score estimation near noise, assuming lower complexity, while the Data-Easy schedule allocates more parameters. Using these schedules, we sample images accordingly, as depicted in Figure 6. In Figure 4, we find that Noise-Easy schedule shows a faster and more stable generation of the butterfly image compared to Data-Easy schedule. More specifically, in the Figure 5, it is evident that the Data-Easy schedule faces challenges in achieving convergence, whereas the Noise-Easy schedule tends to converge successfully, as indicated by the FID value of the baseline model. We thus can conclude that the Noise-Easy schedule, combined with an early exiting scheme, is effective in skipping subsets of parameters in diffusion models.

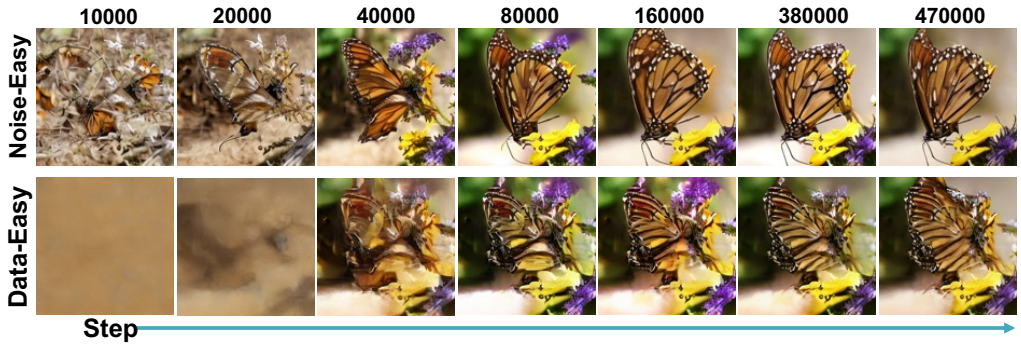

Figure 4. Snapshot samples of Noise-Easy (top) / Data-Easy (bottom) schedules when fine-tuned DiT on ImageNet. Starting from the leftmost column, the images in each column are sampled from progressively increasing fine-tuning steps. While Data-Easy schedule struggles to generate the image of a butterfly, Noise-Easy schedule produces the shape of a butterfly from the 40,000th step onwards.

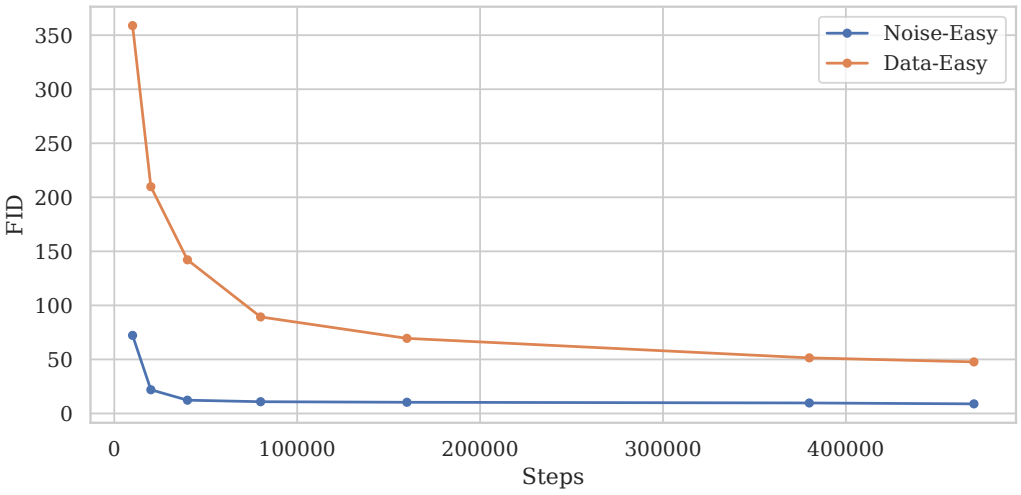

Figure 5. FID metric on the Noise-Easy / Data-Easy schedules fine-tuned DiT on ImageNet. The Noise-Easy schedule aligns with our hypothesis, while the Data-Easy schedule represents the opposite effect.

A.2. Diverse Time-dependent Dropping Schedules

In the Section 2.2, we briefly introduce the concept of time-dependent dropping schedule. We hereby provide the formal definition of D-n schedules. We refer the reader to Table 2. First, the sampling time $[0, 1]$ is divided into five intervals with equal length.

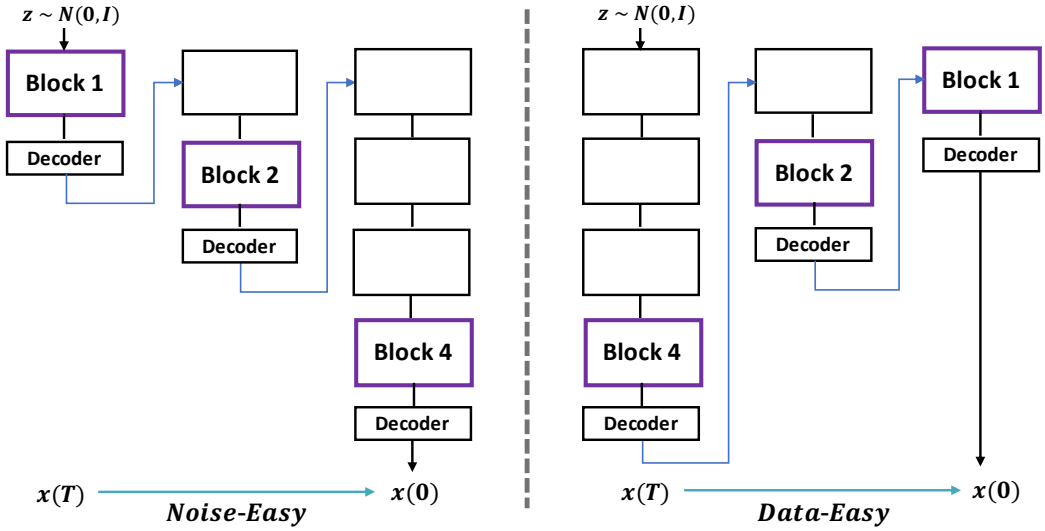

Figure 6. Schematic for sampling procedure based on Noise-Easy (left) / Data-Easy (right) schedules. Noisy-Easy schedule employs fewer blocks near the noise, i.e. earlier sampling steps, while Data-Easy schedule takes the opposite approach.

Table 2. Number of blocks used for varying dropping schedules. All schedules use the same number of blocks within a fixed time interval, except for D6-DiT. Schedule D6-DiT employs a finer-grained dropping schedule based on sampling timestep t. Of note, n in D-n schedule represents the acceleration scale. For instance, D3-DiT and D3-U-ViT schedules bring similar scales in terms of acceleration in sampling speed. Reported acceleration performance is measured with DDPM and EM solver applied to DiT and U-ViT, respectively.

Schedule	Acceleration	Sampling timestep t				
		$[0, 0.2]$	$[0.2, 0.4]$	$[0.4, 0.6]$	$[0.6, 0.8]$	$[0.8, 1.0]$
D1-DiT	16.63%	28	26	24	22	20
D3-DiT	23.43%	28	25	22	19	16
D4-DiT	30.46%	28	24	20	16	12
D6-DiT	38.32%	28	24, 22	20, 18	12, 10	8, 6
D1-U-ViT	16.26%	6	5	4	3	2
D2-U-ViT	22.27%	6	5	3	2	1
D3-U-ViT	24.77%	6	4	3	2	1
D4-U-ViT	28.03%	5	4	2	2	1
D5-U-ViT	30.71%	5	4	2	1	1
D6-U-ViT	32.09%	5	3	2	1	1

For the DiT architecture, we designated the blocks to be dropped among the total of 28 blocks. In the case of D1-DiT, we utilized all 28 blocks near the data. As we moved towards the noise side, we gradually discarded 2 blocks per interval, resulting in a final configuration of using 20 blocks near the noise. The higher the number following 'D', the greater the amount of discarded blocks, thereby reducing the processing time of the diffusion model. For the most accelerated configuration, D4-DiT, we designed a schedule where only 6 blocks pass near the noise. Notably, D6 schedule employs finer-grained time intervals as shown in Table 2. For example, D6-DiT uses 24 blocks if $t \in [0.2, 0.3]$ and 22 blocks if $t \in [0.3, 0.4]$.

For the U-ViT architecture, we aimed to preserve the residual connections by discarding sub-blocks other than nn.Linear, rather than skipping the entire building block. Additionally, the target of dropping was limited to the decoder part, distinguishing it from DiT. Similarly, for D1-U-ViT, we allowed the entire decoder consisting of 6 blocks to pass near the data, and as we moved towards the noise side, we gradually discarded a single block per interval, resulting in only 2 blocks passing near the noise, while the remaining blocks only passed through nn.Linear.

A.3. How to Fine-tune Diffusion Models with Time-dependent Exit Schedule?

When provided with a pre-defined early exit schedule, fine-tuning the pre-trained model becomes necessary. To address this issue, we propose a novel fine-tuning algorithm that focuses on updating minimal information near time $t \to 0$ while updating unseen information near time $t \to 1$. To force the differential information update, we leverage two different techniques: Exponential Moving Average (EMA) and weighted coefficient of $\lambda(t)$.

Algorithm 1 Adjusting the output of intermediate building block of diffusion models

Require: Training dataset $\mathcal{D}$, Teacher parameter $\theta_T = [\theta_T^1, \ldots, \theta_T^N]$, Student parameter $\theta_S = [\theta_S^1, \ldots, \theta_S^N]$, EMA rate α, Pre-defined Exit Schedule $S(t)$, Time-dependent coefficient $\lambda(t)$, Re-weighting cycle C, Learning rate η.

$\theta_T \leftarrow \theta_S, t \sim [0, 1]$ ▷ Initialize the EMA update.
while not converged **do**
  Sample a mini-batch $\mathcal{B} \sim \mathcal{D}$.
  for $i = 1, \ldots, |\mathcal{B}|$ **do**
    Take the input $\boldsymbol{x}_i$ from $\mathcal{B}$.
    for $l = 1, \ldots, N$ **do**
      if $l \leq S(t)$ **then**
        $\tilde{\boldsymbol{x}}_i \leftarrow \text{perturb}(\boldsymbol{x}_i, t)$
        $\ell_i \leftarrow \lambda(t) \cdot \text{loss}(\tilde{\boldsymbol{x}}_i, t)$ ▷ Multiply Time-dependent coefficient
      else
        Break for loop
      end if
    end for
  end for
  $\theta_S \leftarrow \theta_S - \eta \nabla_{\theta_S} \frac{1}{|\mathcal{B}|} \sum_i \ell_i$.
  if epoch $>$ C **then**
    $\lambda(t) \leftarrow 1$ ▷ Re-scale Time-dependent coefficient
  end if
  Update $\theta_T \leftarrow \alpha\theta_T + (1 - \alpha)\theta_S$ ▷ Update the teacher parameter.
end while

B. Experiments

B.1. Setting

Experimental details We evaluate our method across the two models: DiT XL/2 trained on ImageNet with a resolution of 256×256, and U-ViT-S/4 trained on CelebA with a resolution of 64×64. During the fine-tuning step, we employ AdamW (Loshchilov & Hutter, 2017) optimizer for both models. The DiT model was fine-tuned with a constant learning rate of $2 \cdot 10^{-5}$, while cosine annealing learning rate scheduling was employed for the U-ViT model to ensure training stability. For the ImageNet experiments, we used a batch size of 64 and 32 for fine-tuning and sampling, respectively. In the case of U-ViT, the batch sizes were set to 128 for fine-tuning and 250 for sampling. To fine-tune the diffusion models, we utilized a hybrid loss (Nichol & Dhariwal, 2021) with re-weighted time coefficient and linear schedule for injecting noise. Unless stated otherwise, we used $T = 1000$ timesteps for the forward diffusion process.

Evaluation Metrics To validate the effectiveness of our method, we adopt the widely used FID metric (Heusel et al., 2017) for evaluating image generation quality. To mitigate the computational overhead of sampling, we generate 5000 images from our fine-tuned model with a diverse set of samplers. Subsequently, we compute the FID score between the generated samples and the training dataset. Finally, we average the time taken to generate a batch on a single NVIDIA GeForce RTX 3090 GPU in order to check the wallclock time of sampling speed.

Baselines To the best of our knowledge, Token Merging (ToMe; Bolya & Hoffman, 2023) is the only existing work that specifically focuses on reducing the processing time of diffusion models. To ensure a fair comparison with our work, we apply ToMe method to both the DiT and U-ViT backbones, with diverse merging schedules. In the case of stable

diffusion (Rombach et al., 2022), ToMe was integrated into three different modules of the U-Net architecture: self-attention, cross-attention, and MLP. However, given that both DiT and U-ViT are transformer-based architectures, we specifically applied the token merging technique to the self-attention module.

B.2. Additional Experiments

In this section, we provide in-depth experimental results, including both qualitative and quantitative analyses. Firstly, we show the results of repeated experiments in both Table 3 and Table 4, emphasizing the absence of selective or biased results. Additionally, we provide visual representations of randomly generated images for each time-dependent early exiting schedule. In the Figure 7, it illustrates the results obtained by sampling from DiT checkpoint fine-tuned with the D4 schedule using both the DDPM and DDIM sampler. Furthermore, in the Figure 8, it exhibits the results obtained by sampling from U-ViT checkpoint fine-tuned with the D6 schedule using both the EM and DPM sampler.

Table 3. FID evaluation for varying combinations of dropping schedules and sampling methods with DiT on ImageNet 256. Reported values are averaged over five random seeds.

Methods	DDPM 250		DDIM 200		DDIM 100		DDIM 50	
	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)
DiT	9.171 ± 0.064	-	8.937 ± 0.109	-	9.100 ± 0.112	-	9.114 ± 0.055	-
D1-DiT	8.875 ± 0.135	16.63%	8.795 ± 0.167	14.42%	8.889 ± 0.125	14.38%	8.961 ± 0.025	14.37%
D3-DiT	8.773 ± 0.106	23.43%	8.952 ± 0.101	21.04%	8.990 ± 0.166	20.99%	9.042 ± 0.085	20.93%
D4-DiT	8.679 ± 0.070	30.46%	9.126 ± 0.170	28.61%	9.185 ± 0.059	28.70%	9.308 ± 0.073	28.67%
D6-DiT	9.683 ± 0.079	38.32%	11.427 ± 0.190	37.25%	11.414 ± 0.153	36.80%	11.618 ± 0.136	36.71%

Table 4. FID evaluation for varying combinations of dropping schedules and sampling methods with U-ViT on CelebA 64. Reported values are averaged over five random seeds.

Methods	EM 1000		DPM 50		DPM 25	
	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)
U-ViT	4.874 ± 0.044	-	5.520 ± 0.198	-	5.590 ± 0.211	-
D1-U-ViT	4.692 ± 0.050	16.26%	4.8236 ± 0.075	11.07%	4.732 ± 0.076	12.16%
D2-U-ViT	4.817 ± 0.043	22.27%	5.0688 ± 0.079	14.82%	4.919 ± 0.071	16.41%
D3-U-ViT	4.775 ± 0.041	24.77%	4.9448 ± 0.070	15.72%	4.773 ± 0.065	17.60%
D4-U-ViT	4.772 ± 0.058	28.03%	5.1270 ± 0.064	21.02%	4.822 ± 0.047	22.45%
D5-U-ViT	4.746 ± 0.089	30.71%	5.1262 ± 0.057	23.03%	4.911 ± 0.057	24.27%
D6-U-ViT	4.986 ± 0.082	32.09%	5.3268 ± 0.045	23.70%	5.148 ± 0.047	25.67%

B.3. Analysis on ToMe

In this section, we conducted experiments on three different cases for applying ToMe to the building block of a given architecture. The 'F' schedule denotes applying ToMe starting from the front-most block, the 'R' schedule denotes starting from the back-most block, and the 'B' schedule represents symmetric application from both ends. To incorporate diverse merging schedules into both DiT and U-ViT, we implemented ToMe considering the characteristics of each architecture. In the case of DiT, ToMe was directly applied due to its similarity to the existing architecture. However, we carefully determine the specific modules in U-ViT where ToMe should be applied. Since U-ViT treats time and condition as tokens alongside image patches, we exclude these tokens and focus on merging tokens associated with image patches.

Firstly, we provide the experimental results of DiT combined with ToMe in the Table 5 and Table 6. Second, we also present the results of diverse merging schedule applied on the U-ViT, as displayed in the Table 7 and Table 8. Regarding DiT, we have observed that the 'B' schedule proves to be the most effective strategy when combined with ToMe. Conversely, in U-ViT, the 'R' schedule demonstrates the most competitive performance.

Table 5. B merging schedule experiments on DiT with DDIM solver.

DDIM-50	B2		B4		B6		B8		All	
	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)
attn-ratio-2-down-1	9.172	0.29%	9.421	0.37%	10.43	0.60%	13.926	0.69%	117.194	1.92%
attn-ratio-3-down-1	9.313	0.49%	9.745	0.82%	12.918	1.03%	22.495	1.45%	170.170	6.08%
attn-ratio-4-down-1	9.409	0.85%	10.314	1.59%	17.567	2.27%	37.763	2.97%	214.759	10.34%
attn-ratio-5-down-1	9.741	0.91%	11.284	2.26%	25.675	2.63%	58.550	4.07%	247.608	16.66%
attn-ratio-6-down-1	10.014	0.99%	12.441	2.34%	38.124	3.72%	81.987	5.07%	274.591	21.55%

Table 6. R merging schedule experiments on DiT with DDIM solver.

DDIM-50	R2		R4		R6		R8		R10	
	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)
attn-ratio-2-down-1	9.408	0.27%	9.659	0.41%	10.386	0.65%	10.436	0.69%	11.696	0.84%
attn-ratio-3-down-1	9.591	0.47%	10.455	0.92%	12.032	1.10%	12.202	1.30%	15.769	1.67%
attn-ratio-4-down-1	10.384	0.83%	11.724	1.59%	15.270	2.23%	15.937	2.40%	23.510	2.69%
attn-ratio-5-down-1	11.049	0.91%	13.764	1.89%	20.392	2.79%	22.588	3.32%	36.552	4.35%
attn-ratio-6-down-1	12.398	0.99%	17.383	2.34%	30.007	3.72%	34.078	4.39%	57.408	5.88%

Table 7. R merging schedule experiments on U-ViT with EM solver.

EM-1000	R2		R3		R4		R5		All	
	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)
attn-ratio-2-down-1	4.963	-1.95%	6.086	-2.27%	9.890	-4.61%	13.508	-5.71%	342.513	-17.22%
attn-ratio-3-down-1	5.643	0.63%	8.934	-0.49%	18.627	-1.86%	27.984	-3.34%	384.191	-12.75%
attn-ratio-4-down-1	6.991	1.81%	13.743	0.64%	32.183	-0.25%	52.379	-1.06%	424.751	-7.09%
attn-ratio-5-down-1	9.240	2.95%	21.429	2.19%	53.484	1.73%	100.843	1.50%	424.387	-1.17%
attn-ratio-6-down-1	13.048	3.82%	33.521	4.12%	93.509	4.43%	189.431	4.77%	419.851	6.91%

Table 8. R merging schedule experiments on U-ViT with DPM solver.

DPM-25	R2		R3		R4		R5	
	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)
attn-ratio-2-down-1	53.719	-4.21%	63.659	-6.06%	90.006	-7.78%	106.969	-9.21%
attn-ratio-3-down-1	182.380	-3.33%	204.843	-4.76%	259.479	-6.02%	281.183	-7.18%
attn-ratio-4-down-1	298.594	-2.26%	305.796	-3.02%	320.153	-3.86%	326.211	-4.74%
attn-ratio-5-down-1	323.202	-1.29%	335.627	-1.68%	353.834	-2.07%	363.732	-2.40%
attn-ratio-6-down-1	371.790	-0.10%	383.952	0.25%	393.685	0.48%	400.690	0.53%

C. Related Work

Fast Sampling of Diffusion Models Diffusion probabilistic models (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020; Dhariwal & Nichol, 2021) have shown their effectiveness in modeling data distributions and have achieved the state-of-the-art performance, especially in the field of image synthesis. These models employ a progressive denoising approach for noisy inputs which unfortunately leads to heavy computational costs. To overcome this issue, multiple works have been proposed for fast sampling. DDIM (Nichol & Dhariwal, 2021) accelerates the sampling process by leveraging non-Markovian diffusion processes. FastDPM (Kong & Ping, 2021) uses a bijective mapping between continuous diffusion steps and noises. DPM-Solver (Lu et al., 2022) analytically solves linear part exactly while approximating the non-linear part using high-order solvers. DEIS (Zhang & Chen, 2023) utilizes exponential integrator and polynomial extrapolation to reduce discretization errors. In addition to utilizing a better solver, alternative approaches have been proposed, which involve training a student model using network distillation (Salimans & Ho, 2022) or token merging (Bolya & Hoffman, 2023). Our work is orthogonal to these existing approaches, as we focus on reducing the number of processed blocks for each time step, rather than targeting a reduction in the number of sampling steps.

Architecture Design of Diffusion Models The pioneering diffusion models (Ho et al., 2020; Song & Ermon, 2019; Dhariwal & Nichol, 2021), especially in the field of image synthesis, have adopted a U-Net (Ronneberger et al., 2015) backbone architecture with additional modifications including the incorporation of cross- and self-attention layers. Motivated by the recent success of transformer (Vaswani et al., 2017) networks in diverse domains (Brown et al., 2020; Devlin et al., 2019; Xie et al., 2021; Strudel et al., 2021; Liu et al., 2022), several studies have attempted to leverage the Vision Transformer (ViT) (Dosovitskiy et al., 2021) architecture for diffusion models. Gen-ViT (Yang et al., 2022) is a pioneering work that shows that standard ViT can be used for diffusion backbone. U-ViT (Bao et al., 2022) enhances ViT's performance by adding long skip connections and additional convolutional operation. Diffusion Transformers (DiTs) (Peebles & Xie, 2022) investigate the scalability of transformers for diffusion models and demonstrate that larger models consistently exhibit improved performance, albeit at the cost of higher GFLOPs. Our approach focuses on enhancing the efficiency of the transformer through adaptive block selection during calculations, and can be applied to existing transformer-based approaches, such as DiTs, to further optimize their performance.

Early Exiting Scheme for Language Modeling The recent adoption of Large Language Models (LLMs) has brought about significant computational costs, prompting interest in reducing unnecessary computations. Among the various strategies, an early-exiting scheme that dynamically selects computation layers based on inputs has emerged for Transformer-based LLMs. DynaBERT (Hou et al., 2020) transfers knowledge from a teacher network to a student network, allowing for flexible adjustments to the width and depth. Yijin et al. (Liu et al., 2021) employ mutual information and reconstruction loss to assess the difficulty of input words. CAT (Schuster et al., 2021) incorporates an additional classifier that predicts when to perform an early exit. CALM (Schuster et al., 2022) constrains the per-token exit decisions to maintain the global sequence-level meaning by calibrating the early-exiting LLM using semantic-level similarity metrics. Motivated by the aforementioned works, we propose a distinct early-exiting scheme specifically designed for diffusion models.

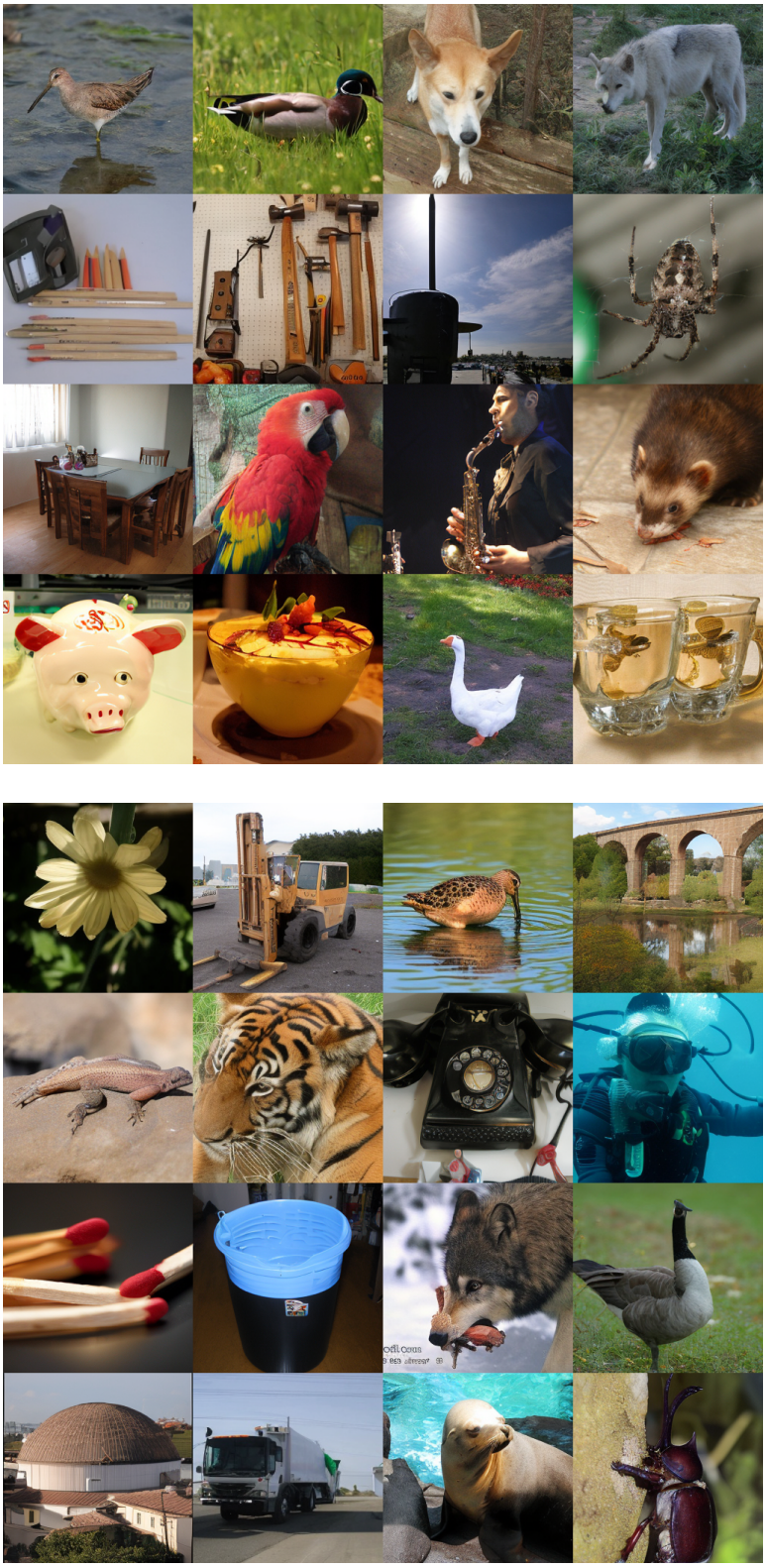

Figure 7. Images sampled from the DiT model fine-tuned with D4-DiT schedule. Top: DDPM sampler-250 steps; Bottom: DDIM sampler-50 steps.

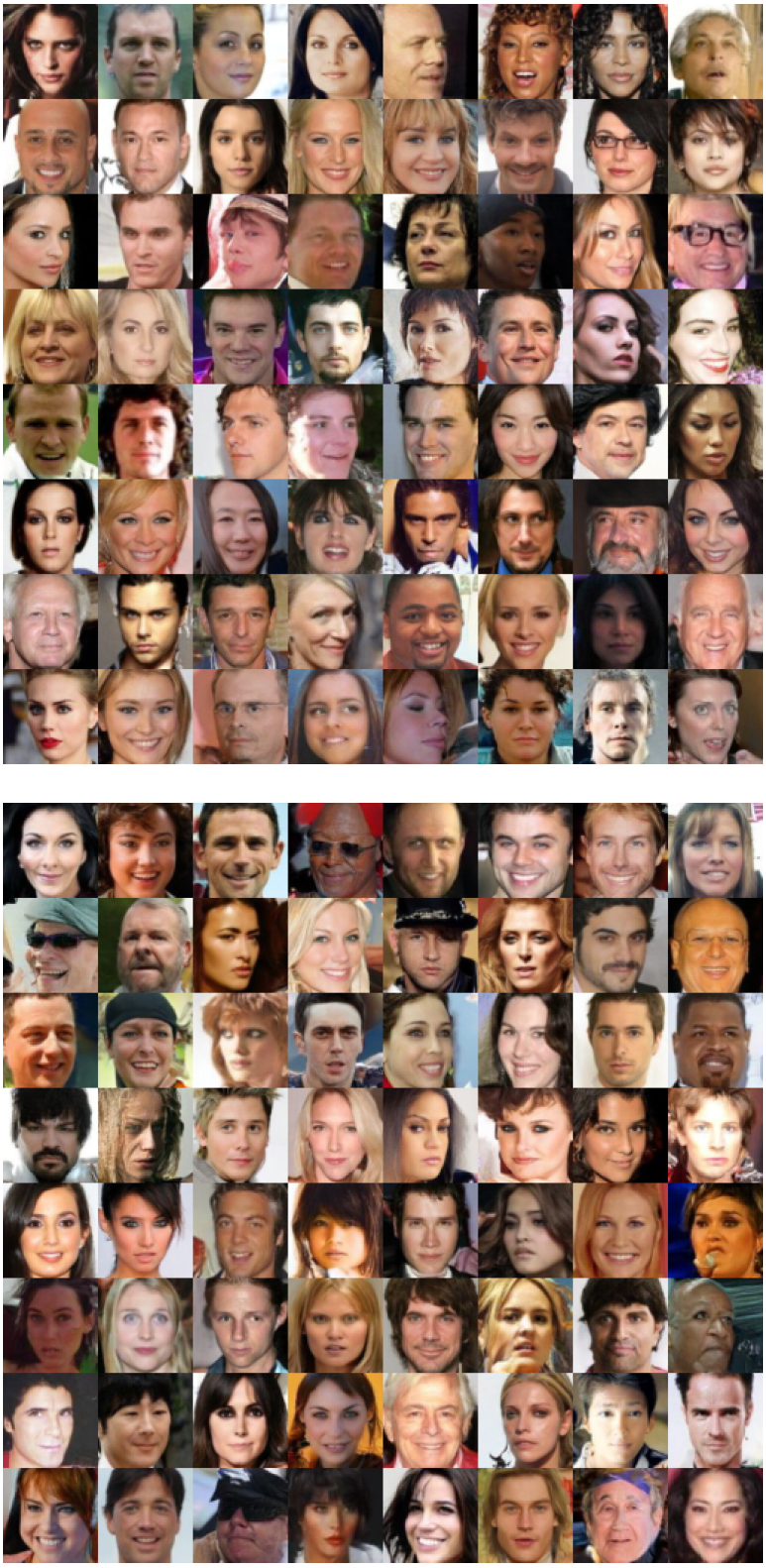

Figure 8. Images sampled from the U-ViT model fine-tuned with D6-U-ViT schedule. Top: EM solver-1000 steps; Bottom: DPM solver-25 steps.