A Simple Early Exiting Framework for Accelerated Sampling in Diffusion Models

Taehong Moon [1] **Moonseok Choi** [2] **EungGu Yun** [3] **Jongmin Yoon** [2] **Gayoung Lee** [4] **Jaewoong Cho** [1] **Juho Lee** [2 5]

Abstract

Diffusion models have shown remarkable performance in generation problems over various domains including images, videos, text, and audio. A practical bottleneck of diffusion models is their sampling speed, due to the repeated evaluation of score estimation networks during the inference. In this work, we propose a novel framework capable of adaptively allocating compute required for the score estimation, thereby reducing the overall sampling time of diffusion models. We observe that the amount of computation required for the score estimation may vary along the time step for which the score is estimated. Based on this observation, we propose an early-exiting scheme, where we skip the subset of parameters in the score estimation network during the inference, based on a time-dependent exit schedule. Using the diffusion models for image synthesis, we show that our method could significantly improve the sampling throughput of the diffusion models without compromising image quality. Furthermore, we also demonstrate that our method seamlessly integrates with various types of solvers for faster sampling, capitalizing on their compatibility to enhance overall efficiency.

1. Introduction

Diffusion probabilistic models (Sohl-Dickstein et al., 2015; Ho et al., 2020) have shown remarkable success in diverse domains including image synthesis (Ho et al., 2020; Dhariwal & Nichol, 2021; Ho et al., 2022a), text-to-image generation (Ramesh et al., 2022; Rombach et al., 2022), 3D point cloud generation (Luo & Hu, 2021), text-to-speech generation (Jeong et al., 2021), and video generation (Ho et al., 2022b). These models learn the reverse process of introducing noise into the data to data and denoise inputs progressively during inference using the learned reverse model.

One major drawback of diffusion models is their slow sampling speed, as they require multiple steps of forward passes through score estimation networks to generate a single sample, unlike the other methods such as GANs (Goodfellow et al., 2014) that require only a single forward pass through a generator network. To address this issue, several approaches have been proposed to reduce the number of steps required for the sampling of diffusion models, for instance, by improving ODE/SDE solvers (Kong & Ping, 2021; Lu et al., 2022; Zhang & Chen, 2023) or distilling into models requiring less number of sampling steps (Salimans & Ho, 2022; Song et al., 2023). Moreover, in accordance with the recent trend reflecting scaling laws of large models over various domains, diffusion models with a large number of parameters are quickly becoming mainstream as they are reported to produce high-quality samples (Peebles & Xie, 2022). Running such large diffusion models for multiple sampling steps incurs significant computational overhead, necessitating further research to optimize calculations and efficiently allocate resources.

On the other hand, recent reports have highlighted the effectiveness of early-exiting schemes in reducing computational costs for Large Language Models (LLMs) (Schuster et al., 2022; Hou et al., 2020; Liu et al., 2021; Schuster et al., 2021). The concept behind early-exiting is to bypass the computation of transformer blocks when dealing with relatively simple or confident words. Given that modern score-estimation networks employed in diffusion models share architectural similarities with LLMs, it is reasonable to introduce the early-exiting idea to diffusion models as well, with the aim of accelerating the sampling speed.

In this paper, we introduce Adaptive Score Estimation (ASE) for faster sampling from diffusion models, drawing inspiration from the early-exiting schemes utilized in LLMs. What sets diffusion models apart and distinguishes our proposal from a straightforward application of the

This work is partially done at KAIST AI. [1]KRAFTON [2]Graduate School of AI, KAIST [3]Independent researcher [4]Naver AI Lab, South Korea [5]AITRICS, South Korea. Correspondence to: Juho Lee <juholee@kaist.ac.kr>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

early-exiting scheme is the time-dependent nature of the score estimation involved in the sampling process. 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, we gain the ability to dynamically control the computation time during the sampling procedure. To accomplish this, we present a time-varying block-dropping schedule and a straightforward algorithm for fine-tuning a given diffusion model to be optimized for this schedule. ASE successfully accelerates the sampling speed of diffusion models while maintaining high-quality samples. Furthermore, ASE is highly versatile, as it can be applied to score estimation networks with various backbone architectures and can be combined with different solvers to further enhance sampling speed. We demonstrate the effectiveness of our method through experiments on real-world image synthesis tasks.

2. 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 lead 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). Recently, consistency model (Song et al., 2023; Song & Dhariwal, 2024) proposed a distillation scheme to directly find the consistency function from the data point within the trajectory of the probability flow. And Kim et al. (2023) refined the consistency model with input-output time parameterization within the score function and adversarial training. While previous approaches focused on reducing the timestep of sampling, recent studies proposed an alternative way to accelerate sampling speed by reducing the processing time of diffusion model itself. In particular, Block Caching (Wimbauer et al., 2023) aim to re-use the intermediate feature which is already computed in previous timestep while Token Merging (Bolya & Hoffman, 2023) target to reduce the number of tokens. Concurrent work (Tang et al., 2023) suggests early exiting scheme on diffusion models. However, it requires additional module which is used to estimate an uncertainty of intermediate features. 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.

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.

3. Method

This section describes our main contribution - Adaptive Score Estimation (ASE) for diffusion models. The section is organized as follows. We first give a brief recap on how to train a diffusion model and provide our intuition on the time-varying complexity of score estimation. Drawing from such intuition, we empirically demonstrate that precise score estimation can be achieved with fewer parameters within a specific time interval. To this end, we present our early-exiting algorithm which boosts inference speed while preserving the generation quality.

3.1. 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

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

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

$$p_\theta(x_{1:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \quad q(x_T|x_0) \approx \mathcal{N}(0, I). \tag{2}$$

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,

$$\begin{aligned}\mathcal{L}(\theta) &= -\sum_{t=1}^{T} \mathbb{E}_q \left[D_{\mathrm{KL}}[q(x_{t-1}|x_t, x_0)\|p_\theta(x_{t-1}|x_t)]\right] \\ &\geq -\log p_\theta(x_0),\end{aligned} \tag{3}$$

where

$$\begin{aligned}q(x_{t-1}|x_t, x_0) &= \mathcal{N}\left(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I\right), \\ \tilde{\mu}_t(x_t, x_0) &= \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_t\right).\end{aligned} \tag{4}$$

The model distribution $p_\theta(x_{t-1}|x_t)$ is chosen as a Gaussian,

$$\begin{aligned}p_\theta(x_{t-1}|x_t) &= \mathcal{N}(x_{t-1}|\mu_\theta(x_t, t), \sigma_t^2 I), \\ \mu_\theta(x_t, t) &= \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(x_t, t)\right),\end{aligned} \tag{5}$$

and the above loss function then simplifies to

$$\mathcal{L}(\theta) = \sum_{t=1}^{T} \mathbb{E}_{x_0, \varepsilon_t}\left[\lambda(t)\left\|\varepsilon_t - \varepsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon_t, t)\right\|^2\right], \tag{6}$$

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.

Under a simple reparameterization, one can also see that,

$$\nabla_{x_t}\log q(x_t|x_0) = -\frac{\varepsilon_t}{\sqrt{1-\bar{\alpha}_t}} \approx -\frac{\varepsilon_\theta(x_t, t)}{\sqrt{1-\bar{\alpha}_t}} := s_\theta(x_t, t), \tag{7}$$

where $s_\theta(x_t, t)$ is the score estimation network. In this parameterization, the loss function can be written as,

$$\mathcal{L}(\theta) = \sum_{t=1}^{T} \mathbb{E}_{x_0, x_t}\left[\lambda_t'\|\nabla_{x_t}\log q(x_t|x_0) - s_\theta(x_t, t)\|^2\right], \tag{8}$$

so learning a diffusion model amounts to regressing the score function of the distribution $q(x_t|x_0)$. 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 $\mathbb{E}_{x_0|x_t}\left[\nabla_{x_t}\log q(x_t|x_0)\right] = \nabla_{x_t}\log q(x_t)$.

Suppose we train our diffusion model using the standard parameterization (i.e., ε-parameterization), where the objective is to minimize the gap $\|\varepsilon_\theta - \varepsilon\|^2$. When t is close to 1, this gap primarily represents noise, constituting only a small fraction of the entire x_0. Consequently, it indicates that learning does not effectively occur in the proximity to the noise. Given that a diffusion model is trained across all time steps with a single neural network, it is reasonable to anticipate that a significant portion of the parameters are allocated for the prediction of near data regime (t close to 0). This intuition leads to our dropping schedule pruning more parameters when t is close to 1.

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:

$$x_{t+1} = x_t + \beta\nabla_x \log p(x_t) + \sqrt{2\beta}z_t, \tag{9}$$

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 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. Our experiments indicate that it's feasible to reduce τ by performing score estimation for specific time intervals using fewer parameters. While one could suggest employing differently sized models for estimating scores at various time intervals to reduce overall sampling time, our strategy introduces a simple early exiting framework within a single model, avoiding extra memory consumption. Furthermore, 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 near prior distribution.

3.2. Adaptive Layer Usage in Diffusion Process

We hereby introduce an early exiting framework to accelerate the sampling process of pre-trained diffusion models. Drawing upon the intuition presented in § 3.1, we first explain how to decide the amount of parameters to be used for score estimation. After dropping the selected blocks, we design a fine-tuning algorithm to adjust the output of

Figure 1. Snapshot samples of Noise-Easy / Data-Easy schedules when fine-tuned DiT on ImageNet. While the data-easy schedule struggles to produce a discernible dog image, the noise-easy schedule successfully generates a clear dog image, achieving a converged FID score of 8.88.

intermediate building blocks of diffusion models.

Which time interval can be accurately estimated with fewer parameters? To validate our hypothesis in the context of training diffusion models, we conduct a toy experiment regarding the difficulty of score estimation for different time steps. We conduct tests under two scenarios: one assuming that estimation near the prior distribution requires fewer parameters (Noise-Easy schedule), and the other assuming that estimation near the data distribution demands fewer parameters (Data-Easy schedule). As shown in Figure 1, one can easily find that the noise-easy schedule successfully generates a clear dog image where as the data-easy schedule struggles to produce a discernible dog image.

Which layer can be skipped for score estimation? To accelerate inference in diffusion models, we implement a dropping schedule that takes into account the complexity of score estimation near $t \to 1$ compared to $t \to 0$. For the DiT model trained on ImageNet, which consists of 28 blocks, we design a dropping schedule that starts from the final block. Based on our intuition, we drop more DiT blocks as time approaches 1, as shown in Figure 2. Conversely, for scores near the data, which represent more challenging tasks, we retain all DiT blocks to utilize the entire parameter set effectively.

In U-ViT, the dropping schedule has two main distinctions from DiT: the selection of candidate modules to drop and the subset of parameters to be skipped. Unlike DiT, we limit dropping to the decoder part in U-ViT. This decision is 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 preserve the linear layer of a building block to retain feature information connected through skip connections, while skipping the remaining parameters.

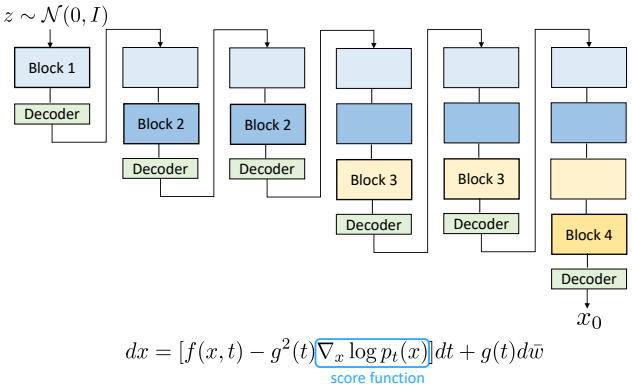

Figure 2. Schematic for time-dependent exit schedule. Considering the varying difficulty of score estimation, we drop more building blocks of architecture near noise. While we skip the whole building blocks in DiT, we partially skip the blocks in U-ViT due to the long skip-connection.

3.3. Fine-tuning Diffusion Models

Following the removal of blocks based on a predetermined dropping schedule, we need to fine-tune the model. 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 encounters input values that differ from the distribution it learned during its initial training, requiring adjustments.

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: (i) adapting Exponential Moving Average (EMA), and (ii) weighting the coefficients $\lambda(t)$.

The EMA technique is employed to limit the frequency of information updates, thereby preserving the previous knowledge acquired by the model during its initial training phase. A high EMA rate results in a more gradual modification of parameters. In our approach, we deliberately maintain a high EMA rate to enhance the stability of our training process. During the gradual parameter update, we aim to specifically encourage modifications in a subset of parameters that align the predicted scores more closely with the prior distribution. To prioritize the learning of this score distribution, we apply a higher coefficient to the $\lambda(t)$ term, which in turn multiplies on the expectation of the training loss. Once the model's performance appears to have plateaued, we adjust the $\lambda(t)$ value back to 1, aiming to facilitate comprehensive learning across the entire score distribution spectrum. We provide the pseudo-code for fine-tuning diffusion models in Appendix A.

4. Experiments

4.1. Experimental Setting

Experimental Details. Throughout the experiments, we use DiT (Peebles & Xie, 2022) and U-ViT (Bao et al., 2022), the two representative diffusion models. We employ three pre-trained models: (1) DiT XL/2 trained on ImageNet (Krizhevsky et al., 2017) with the resolution of 256×256; (2) U-ViT-S/4 trained on CelebA (Liu et al., 2015) with the resolution of 64×64; (3) PixArt-α-SAM-256 trained on SAM dataset (Kirillov et al., 2023). For the fine-tuning step in both DiT and U-ViT experiments, we employ a hybrid loss (Nichol & Dhariwal, 2021) with a re-weighted time coefficient and linear schedule for injecting noise. We use AdamW (Loshchilov & Hutter, 2017) optimizer with the learning rate of $2 \cdot 10^{-5}$. We use cosine annealing learning rate scheduling to ensure training stability for the U-ViT models. Batch size is set to 64, and 128 for fine-tuning DiT XL/2, U-ViT-S/4, respectively. We use $T = 1000$ time steps for the forward diffusion process. In case of PixArt experiment, we fine-tune our model with 100K SAM data, the batch size of 200×4, and 2200 iterations while the pre-trained model is trained with 10M data, the batch size of 176×64 and 150K iterations. For further experimental details, we refer readers to Appendix A.

Evaluation Metrics. We employ Fréchet inception distance (FID) (Heusel et al., 2017) for evaluating image generation quality of diffusion models. We compute the FID score between 5,000 generated samples from diffusion models and the full training dataset. In case of text-to-image experiment, we measure the FID score with MS-COCO valid dataset (Lin et al., 2014). To evaluate the sampling speed of diffusion models, we report the wall-clock time required to generate a single batch of images on a single NVIDIA A100 GPU.

Baselines. In this study, we benchmark our method against a range of recent techniques which aims reducing the processing time of diffusion models. This includes DeeDiff (Tang et al., 2023), token merging (ToMe; Bolya & Hoffman, 2023), and block caching (Wimbauer et al., 2023). When extending ToMe to U-ViT architecture, we specifically apply the token merging technique to self-attention and MLP modules within each block of the U-ViT. Of note, U-ViT treats both time and condition as tokens in addition to image patches. To improve generative modeling, we exclude these additional tokens and focus solely on merging tokens associated with image patches, following the approach outlined by (Bolya & Hoffman, 2023). For block caching, we employ caching strategies within the attention layers. Naive caching may aggravate feature misalignment especially when caching is more aggressive in order to achieve faster sampling speed. To resolve such an issue, (Wimbauer et al., 2023) further propose shift-scale alignment mechanism. As we explore high-acceleration regime, we report results for both the original block caching technique and its variant with the shift-scale mechanism applied (termed SS in Figure 3). We only report the best performance attained among the diverse hyperparameter settings in the following sections. The remaining results will be deferred to Appendix C as well as experimental details for baseline strategies.

4.2. Inference Speed and Performance Trade-off

Figure 3 presents a trade-off analysis between generation quality and inference speed, comparing our approach to other baseline methods. We can readily find that ASE largely outperforms both ToMe and block caching strategies. ASE boosts sampling speed by approximately 25-30% while preserving the FID score.

Techniques based on feature similarity, such as ToMe and block caching, are straightforward to implement yet fail to bring significant performance gain, or even in some cases, bring an increase in processing time. This can primarily be attributed to the additional computational overhead introduced by token partitioning and the complexity of bipartite soft matching calculations for token merging, which outweighs the advantages gained from reducing the number of tokens. This observation is particularly noteworthy, as even for the CelebA dataset, the number of tokens in U-ViT remains relatively small, and U-ViT does not decrease the token count through layers, as is the case with U-Net.

Regarding block caching, it yields only slight enhancements in inference speed while preserving the quality of generation. Although block caching can be straightforwardly applied to various diffusion models, it encounters a notable constraint: it relies significantly on scale-shift alignment, necessitating extra fine-tuning. Additionally, its effectiveness depends on the specifc architectural characteristics of the model being used. We postulate that this dependency may be related to the presence of residual paths within the architecture. It is crucial to highlight that our method effectively increases sampling speed without sacrificing the quality of the generated output.

In Table 2, we further compare DeeDiff with our method using the performances reported in (Table 1; Tang et al., 2023). ASE and DeeDiff share the same essence as both are grounded in the early-exiting framework. The distinction lies in the dynamic sampling process. To determine when to perform early-exiting for dynamic sampling, an additional module needs to be added to the model, whereas ASE does not require any additional memory. Furthermore, ASE exhibits faster acceleration while maintaining or improving FID, but for DeeDiff, there is a trade-off between the advantage in GFLOPs and the potential disadvantage in

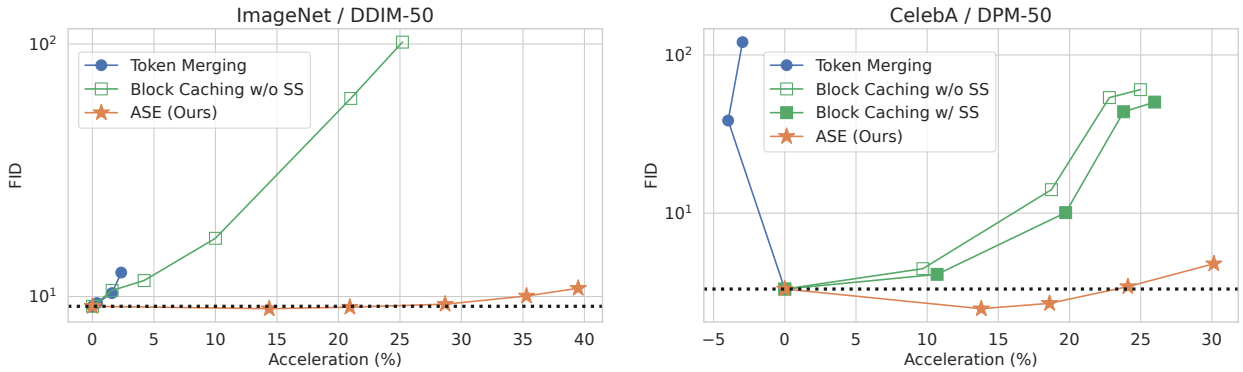

Figure 3. Trade-off between image generation quality and sampling speed on ImageNet with DiT (left) and CelebA with U-ViT (right). We generate samples from DDIM and DPM sampler with 50 steps for ImageNet and CelebA, respectively. ASE largely outperforms other techniques, preserving FID score while boosting sampling speed by approximately 25-30%. Here, SS stands for scale-shift adjustment used together with block caching.

Table 1. Trade-off between image generation quality and sampling speed on ImageNet (DiT; DDPM sampler) and CelebA (U-ViT; EM sampler). ASE consistently maintains image generation quality while achieving a notable increase in sampling speed of approximately 30%; ASE can be effectively used in conjunction with fast solvers. Refer to Table 5 in Appendix A for detailed description of our dropping schedules.

ImageNet (DiT)	DDPM-250	
	FID (↓)	Accel. (↑)
Baseline	9.078	-
D2-DiT	8.662	23.43%
D3-DiT	**8.647**	30.46%
D4-DiT	9.087	34.56%
D7-DiT	9.398	38.92%

CelebA (U-ViT)	EM-1000	
	FID (↓)	Accel. (↑)
Baseline	2.944	-
D1-U-ViT	**2.250**	21.3%
D2-U-ViT	2.255	24.8%
D3-U-ViT	3.217	29.7%
D6-U-ViT	4.379	32.6%

generation quality. In the case of ToMe and block caching, both methods fall significantly short of achieving the performance of ASE or DeeDiff.

4.3. Compatability with Diverse Sampling Solvers

We demonstrate the compatibility of the proposed method with diverse sampling methods. First of all, we verify that our method can be successfully applied to accelerate sampling speed without degrading generation qualtiy. In Table 1, we generated samples with DDPM (Ho et al., 2020) in DiT architecture and get samples from Euler-Maruyama solver. Here, we present results of four varying dropping schedules in each experiments. In a nutshell, 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. We refer readers to Table 5 for detailed guide on ASE dropping schedules.

Furthermore, we show that our method can be seamlessly incorporated with fast sampling solver, such as DDIM (Song et al., 2020) solvers and DPM solver (Lu et al., 2022). From the DiT results presented in , we we observe that our approach effectively achieves faster inference while utilizing fewer parameters, yet maintains the same level of performance. In case of U-ViT, we show that our method notably achieves an over 30% acceleration, while preserving similar quality in generation with the DPM solver. Notably in Figure 4, we highlight that our method is robust across various time steps within both DDIM and DPM solver. This indicates that our method effectively estimates scores across the entire time interval. The reasons for our method's robustness and efficiency in achieving faster inference will be further explained in § 5.

4.4. Large Scale Text-to-Image Generation Task

To demonstrate that our method can be extended to large-scale datasets, we apply it to the pre-trained PixArt-α model. While there may be concerns that fine-tuning with a large-scale dataset could potentially slow down the fine-tuning process, we find that using only 1% of the original data is sufficient for our method to achieve the desired performance. To evaluate our method, we employ a DPM solver with 20 steps and classifier-free guidance (Ho & Sal-

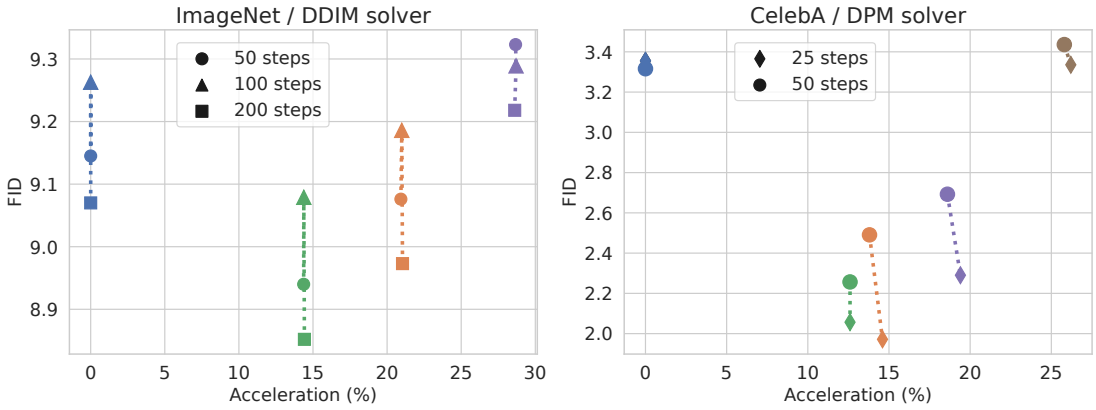

Figure 4. Robustness of ASE across varying sampling timesteps: ImageNet with DDIM solver (left), and CelebA with DPM solver (right). Both experiments employed U-ViT architecture. ASE displays robust performance throughout different timesteps in both different experimental settings.

Table 2. Trade-off between image generation quality and sampling speed on CelebA (U-ViT; DPM-50). Compared to the other baselines, ASE displays a remarkable sampling speed in terms of acceleration in GFLOPs.

Methods	CelebA	
	Accel. (↑)	FID (↓)
U-ViT	-	2.87
DeeDiff (Tang et al., 2023)	45.76%	3.9
ToMe (Bolya & Hoffman, 2023)	3.05%	4.963
Block Caching (Wimbauer et al., 2023)	9.06%	3.955
ASE (Ours)	23.39%	1.92

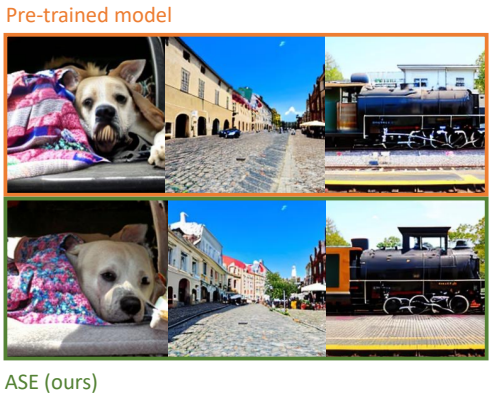

Figure 5. Comparison between samples produced by pre-trained PixArt-α and ASE-enhanced PixArt-α. Text prompts are randomly chosen.

imans, 2022). Although the original model achieves an FID score of 12.483, the ASE-enhanced model attains an FID score of 12.682, with a 14% acceleration in terms of wall-clock time. An example of an image generated from a given prompt is shown in Figure 5.

5. Further Analysis

Ablation Study on Dropping Schedules. Although it is empirically understood that we can eliminate more parameters near the prior distribution, it remains to be determined which time-dependent schedules yield optimal performance in generation tasks. To design an effective dropping schedule, we conduct an ablation study as follows: we create four distinct schedules that maintained the same total amount of parameter dropping across all time intervals, but vary the amount of dropping for each specific interval. These schedules are tested on a U-ViT backbone trained on the CelebA dataset. Specifically, the decoder part of this architecture consists of six blocks, and Figure 6 illustrates how many blocks are utilized at each time t. By fine-tuning in this manner, we evaluate the generation quality of the models, as shown in Table 3. As the results indicate, Schedule 1 outperforms the others, demonstrating the most superior and stable performance across varying time steps.

Viewpoint of Multi-task Learning. Diffusion models can be seen as a form of multi-task learning, as they use a single neural network to estimate the scores at every time t. In the context of multi-task learning, negative transfer phenomenon can occur, leading to a decrease in the generation quality of diffusion models. Recent work, such as DTR (Park et al., 2023), improve generation quality by jointly training a mask with the diffusion model. This approach minimizes negative transfer by reducing interference between tasks. Similarly, our method, despite using fewer parameters, is designed to achieve a comparable effect. By explicitly distinguishing the parameters used for predicting specific intervals through early-exiting, our approach can mitigate the issues associated with negative transfer.

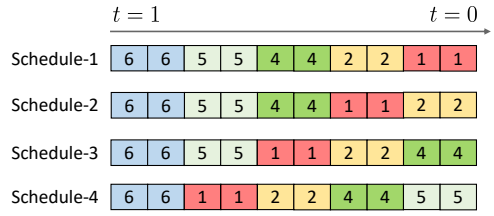

Figure 6. Dropping schedules designed for the ablation study. We divide the sampling time into ten uniform intervals, and drop a specific amount of blocks. The number indicates the amount of blocks left after dropping the rest.

Table 3. FID score on CelebA dataset with U-ViT backbone across ablated dropping schedules. In both DPM-25 and DPM-50, schedule-1 exhibits the best performance.

Methods	DPM-25	DPM-50
Schedule-1	**2.116**	**2.144**
Schedule-2	2.456	2.28
Schedule-3	2.173	3.128
Schedule-4	2.966	3.253

To illustrate the efficacy of our method in mitigating negative transfer, we hereby conduct a toy experiment. Consider score estimation over a specific time interval $t \in [s, l]$ as a single task. In the experiment, we equally divide the whole sampling time into ten intervals, thereby defining a total of ten tasks. To verify the presence of negative transfer in the diffusion model, we create both a baseline model and expert models trained specifically for each interval. In order to check whether the pre-trained model is sufficiently trained, we further train the baseline model, and Table 4 shows that further-training degrades the performance. Also, the multi-experts model outperforms the baseline model, indicating successful reduction of task interference. Furthermore, replacing the pre-trained model with the ASE module (*Mixed-k* models) in a single time interval leads to performance gains. In Table 4, we can readily observe that the mixed schedules outperform the baseline model across all intervals in terms of image generation quality. This finding suggests that our training approach can not only effectively boost sampling speed but also preserves model performance via mitigating negative transfer effect.

6. Conclusion and Limitations

In this paper, we present a novel method that effectively reduces the overall computational workload 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 DiTs models. A 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.

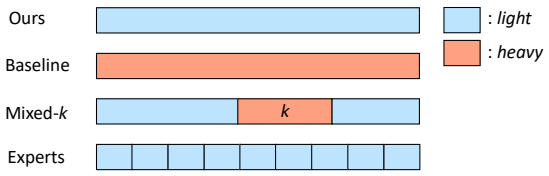

Figure 7. Schematic for different types of dropping schedules designed to validate negative transfer phenomenon. Mixed-k replaces the original heavy model with light ASE model only on kth time interval. Experts employ individually fine-tuned heavy models at each time interval.

Table 4. FID score on CelebA dataset with U-ViT backbone across NTR-inspired dropping schedules. Experts outperform both baseline and further fine-tuned model thereby indicating that negative transfer does exist. Moreover, all the mixed-k schedules, despite only replacing a single time interval, demonstrate improved performance compared to the original baseline model.

Methods	DPM-25	DPM-50
Baseline	3.355	3.316
Further-trained	4.262	4.028
Multi-Experts	**2.987**	**2.942**
Mixed-1	2.938	3.054
Mixed-3	2.654	3.232
Mixed-5	3.287	3.187
Mixed-7	2.292	2.969
Mixed-9	2.933	3.027

Impact Statement

Our work is improving diffusion models which can be misused for generating fake images or videos, contributing to the spread of deepfake content or the creation of misleading information. Also, given that these models are trained on data collected from the internet, there is a risk of harmful biases being embedded in the generated samples such as emphasizing stereotypes.

Acknowledgement

The authors would like to express their sincere gratitude to Jaehyeon Kim and Byeong-Uk Lee for their insightful and constructive discussions. This work was partly supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea govern-

ment(MSIT) (No.RS-2019-II190075 Artificial Intelligence Graduate School Program(KAIST), KAIST-NAVER Hypercreative AI Center, Korea Foundation for Advanced Studies (KFAS), No.2022-0-00713, Meta-learning Applicable to Real-world Problems), and National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF2021M3E5D9025030).

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. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 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. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 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. In *International Conference on Learning Representations (ICLR)*, 2021.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, 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. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 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. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 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.

Kim, D., Lai, C.-H., Liao, W.-H., Murata, N., Takida, Y., Uesaka, T., He, Y., Mitsufuji, Y., and Ermon, S. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *arXiv preprint arXiv:2310.02279*, 2023.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.

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.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

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., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

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. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 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. In *Proceedings of The 38th International Conference on Machine Learning (ICML 2021)*, pp. 8162–8171, 2021.

Park, B., Woo, S., Go, H., Kim, J.-Y., and Kim, C. Denoising task routing for diffusion models. *arXiv preprint arXiv:2310.07138*, 2023.

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. In *International Conference on Learning Representations (ICLR)*, 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. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of The 32nd International Conference on Machine Learning (ICML 2015)*, pp. 2256–2265, 2015.

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

Song, Y. and Dhariwal, P. Improved techniques for training consistency models. *International Conference on Learning Representations (ICLR)*, 2024.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019.

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. Consistency models. In *Proceedings of The 39th International Conference on Machine Learning (ICML 2023)*, 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.

Tang, S., Wang, Y., Ding, C., Liang, Y., Li, Y., and Xu, D. Deediff: Dynamic uncertainty-aware early exiting for accelerating diffusion model generation. *arXiv preprint arXiv:2309.17074*, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017.

Wimbauer, F., Wu, B., Schoenfeld, E., Dai, X., Hou, J., He, Z., Sanakoyeu, A., Zhang, P., Tsai, S., Kohler, J., et al. Cache me if you can: Accelerating diffusion models through block caching. *arXiv preprint arXiv:2312.03209*, 2023.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 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. In *Proceedings of The 39th International Conference on Machine Learning (ICML 2023)*, 2023.

A. Experimental Details

A.1. How to design dropping schedules?

Diverse Time-dependent Dropping Schedules In Table 1, we briefly introduce the difference between the diverse schedules, D1 to D6. We hereby provide the formal definition of D-n schedules. We refer the reader to Table 5. First, the sampling time $[0, 1]$ is divided into ten intervals with equal length.

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 some blocks per interval, resulting in a final configuration of using the smallest number of 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, D7-DiT, we designed a schedule where only 8 blocks pass near the noise.

Table 5. Number of blocks used for varying dropping schedules. All schedules use the same number of blocks within a fixed time interval. 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									
		$[0, 0.1]$	$[0.1, 0.2]$	$[0.2, 0.3]$	$[0.3, 0.4]$	$[0.4, 0.5]$	$[0.5, 0.6]$	$[0.6, 0.7]$	$[0.7, 0.8]$	$[0.8, 0.9]$	$[0.9, 1.0]$
D2-DiT	23.43%	28	28	25	25	22	22	19	19	16	16
D3-DiT	30.46%	28	28	24	24	20	20	16	16	12	12
D4-DiT	34.56%	28	28	26	24	20	18	12	10	8	8
D7-DiT	38.92%	28	28	24	21	18	15	10	10	8	8
D1-U-ViT	21.3%	6	6	4	4	2	2	2	2	1	1
D2-U-ViT	24.8%	5	5	4	4	2	2	1	1	1	1
D3-U-ViT	29.7%	3	3	2	2	2	2	1	1	1	1
D6-U-ViT	32.6%	2	2	2	2	1	1	1	1	1	1

For the U-ViT architecture as we depicted in Figure 8, 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 1 blocks passing near the noise, while the remaining blocks only passed through nn.Linear.

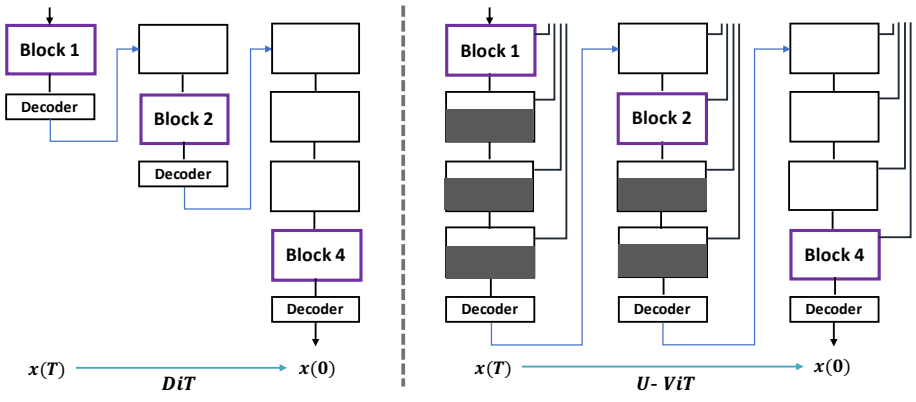

Figure 8. Schematic for the dropping schedules of DiT (left) and U-ViT (right). Due to the existence of residual connections in U-ViT, dropping encoder or decoder blocks in a straightforward manner cause severe performance degradation. In the case of U-ViT, the decoder blocks, except for the linear layer connected to encoder residual connections, are dropped.

A.2. Pseudo-code for fine-tuning diffusion models

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.

$\theta_T \leftarrow \theta_S, t \sim [0, 1]$
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**
        $\tilde{\boldsymbol{x}}_i \leftarrow \text{perturb}(\boldsymbol{x}_i, t)$
        $\ell_i \leftarrow \lambda(t) \cdot \text{loss}(\tilde{\boldsymbol{x}}_i, t)$
      else
        Break for loop
      end if
    end for
  end for
  $\theta_S \leftarrow \theta_S - \eta \nabla_{\theta_S} \frac{1}{|\mathcal{B}|} \sum_i \ell_i$.
  Update $\theta_T \leftarrow \alpha\theta_T + (1 - \alpha)\theta_S$
end while

A.3. Computational Efficiency of ASE

Additional Fine-tuning cost of ASE Compared with ToMe (Bolya & Hoffman, 2023) and Block Caching (Wimbauer et al., 2023), our method requires fine-tuning. Nonetheless, we demonstrate its negligible fine-tuning cost and high efficiency by reporting the computational costs for fine-tuning in Table 6.

Table 6. Fine-tuning costs when we apply ASE into pre-trained DiT on ImageNet and U-ViT on CelebA. These tables show the number of iterations and batch sizes used during the fine-tuning process.

ImageNet (DiT)	iteration * batch size
Baseline	400K * 256
D2-DiT	400K * 32 (12.50%)
D3-DiT	450K * 32 (14.06%)
D4-DiT	500K * 32 (15.63%)

CelebA (U-ViT)	iteration * batch size
Baseline	500K * 128
D1-U-ViT	40K * 128 (8%)
D2-U-ViT	50K * 128 (10%)
D3-U-ViT	150K * 64 (15%)
D6-U-ViT	200K * 64 (20%)

Results on actual inference time of ASE In Table 7, we provide additional results on wall-clock time. We note that the acceleration rate in the original paper is also measured in terms of wall-clock time.

Table 7. Wall-clock time of generating samples with ASE-enhanced models. Left table is the result of DiT model fine-tuned on ImageNet and right table is the result of U-ViT model fine-tuned on CelebA.

ImageNet (DiT)	DDPM-250	
	FID (↓)	Wall-clock time (s) (↓)
Baseline	9.078	59.60
D2-DiT	8.662	45.63
D3-DiT	**8.647**	41.44
D4-DiT	9.087	39.00
D7-DiT	9.398	36.40

CelebA (U-ViT)	EM-1000	
	FID (↓)	Wall-clock time (s) (↓)
Baseline	2.944	216.70
D1-U-ViT	**2.250**	170.54
D2-U-ViT	2.255	162.95
D3-U-ViT	3.217	152.34
D6-U-ViT	4.379	146.05

B. Related Work

Transformers in 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.

C. Further Analysis on Baselines

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. In the Figure 3, we report the experiment results that showed the most competitive outcomes. Furthermore, we present the remaining experiments conducted using various merging schedules, as illustrated in Table 8, Table 9. In summary, for the DiT architecture, the 'B' schedule performed well, while the 'R' schedule demonstrated satisfactory performance for the U-ViT architecture.

Table 8. Diverse merging schedule experiments on DiT with DDIM sampler.

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 9. Diverse merging schedule experiments on U-ViT with DPM sampler.

DPM-50	R2		R3		R4		R5	
	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)
attn-ratio-2-down-1	38.505	-3.98%	45.544	-5.89%	65.755	-7.51%	79.086	-9.15%
attn-ratio-3-down-1	120.596	-2.97%	141.073	-4.53%	200.132	-5.85%	232.040	-7.07%
attn-ratio-4-down-1	264.153	-2.13%	279.270	-2.76%	311.823	-3.69%	319.599	-4.57%
attn-ratio-5-down-1	308.350	-1.13%	315.334	-1.53%	332.565	-1.90%	343.486	-2.02%
attn-ratio-6-down-1	330.501	0.05%	344.353	0.41%	362.002	0.69%	372.612	1.10%

Analysis on Block Caching To ensure fair comparison between baseline methods, we faithfully implement block caching algorithm on both DiT and U-ViT architecture. In this experiment, we applied it to the attention part of the U-ViT blocks, and Table 10 shows the trade-off between generation quality and inference speed depending on the presence or absence of the scale-shift mechanism.

D. Qualitative Comparison

We present comprehensive experimental results, primarily including qualitative analyses. Figure 9 and Figure 10 shows the superior quality of generated samples under various dropping schedules. Additionally, in the Figure 11 and Figure 12, we show the robustness of ASE across varing sampling timesteps. Notably, we provide visual representations of randomly generated images for each time-dependent early exiting schedule. In the Figure 13, it illustrates the results obtained by sampling from fine-tuned DiT checkpoint using both the DDPM and DDIM sampler. Similarly, in the Figure 14, it exhibits the results obtained by sampling from fine-tuned U-ViT checkpoint using both the EM and DPM sampler.

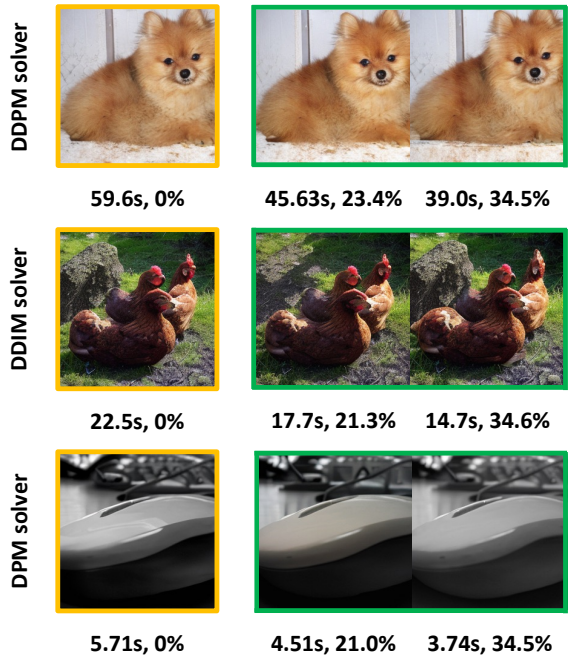

Figure 9. Images sampled from ASE-enhanced DiT model with diverse dropping schedules.

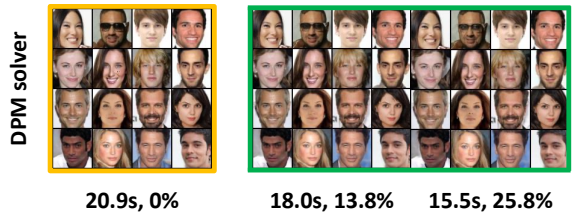

Figure 10. Images sampled from ASE-enhanced U-ViT model with diverse dropping schedules.

Table 10. Additional block caching experiments on U-ViT with DPM sampler.

DPM-50	Attn(wo SS)		Attn(w SS)	
	FID (↓)	Accel. (↑)	FID (↓)	Accel. (↑)
attn-ths-0.1	4.462	9.70%	3.955	9.06%
attn-ths-0.2	14.083	18.73%	9.707	18.11%
attn-ths-0.3	53.770	22.80%	32.518	22.35%
attn-ths-0.4	60.390	24.98%	45.523	24.26%

Figure 11. Images sampled from the fine-tuned DiT model with DPM sampler.

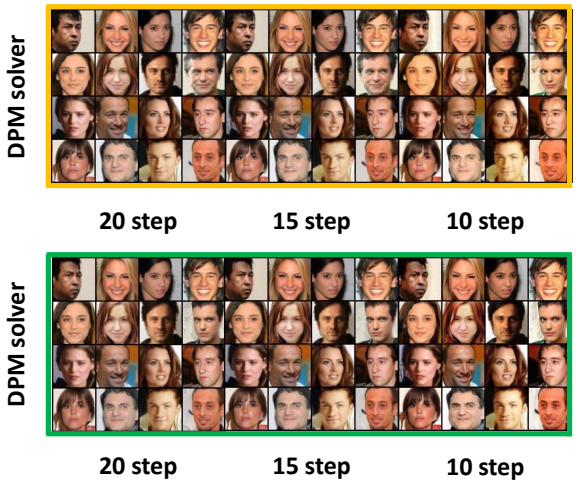

Figure 12. Images sampled from the fine-tuned U-ViT model with DPM sampler.

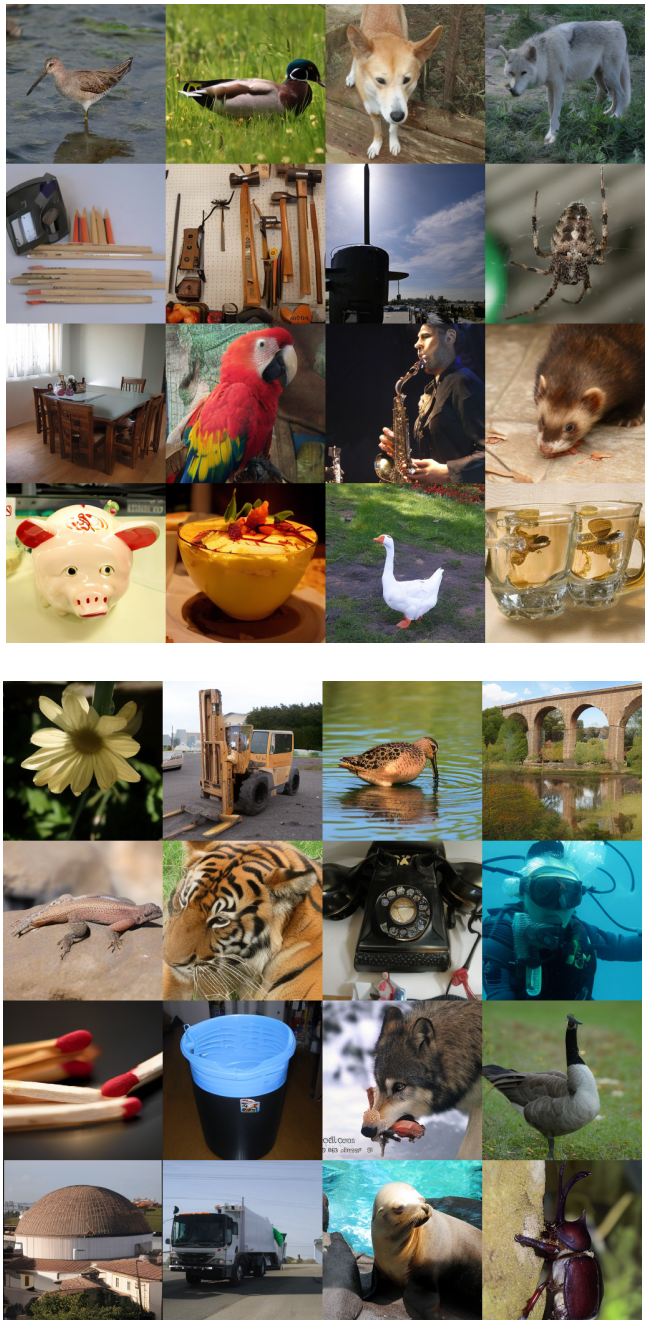

Figure 13. Images sampled from the fine-tuned DiT model. Top: DDPM sampler-250 steps; Bottom: DDIM sampler-50 steps.

Figure 14. Images sampled from the fine-tuned U-ViT model. Top: EM solver-1000 steps; Bottom: DPM solver-25 steps.