FRDiff : Feature Reuse for Universal Training-free Acceleration of Diffusion Models

Junhyuk So[*1], Jungwon Lee[*2], and Eunhyeok Park[1,2]

[1] Department of Computer Science and Engineering,
[2] Graduate School of Artificial Intelligence,
POSTECH, Pohang, South Korea
{junhyukso,leejungwon,eh.park}@postech.ac.kr

Abstract. The substantial computational costs of diffusion models, especially due to the repeated denoising steps necessary for high-quality image generation, present a major obstacle to their widespread adoption. While several studies have attempted to address this issue by reducing the number of score function evaluations (NFE) using advanced ODE solvers without fine-tuning, the decreased number of denoising iterations misses the opportunity to update fine details, resulting in noticeable quality degradation. In our work, we introduce an advanced acceleration technique that leverages the temporal redundancy inherent in diffusion models. Reusing feature maps with high temporal similarity opens up a new opportunity to save computation resources without compromising output quality. To realize the practical benefits of this intuition, we conduct an extensive analysis and propose a novel method, FRDiff. FRDiff is designed to harness the advantages of both reduced NFE and feature reuse, achieving a Pareto frontier that balances fidelity and latency trade-offs in various generative tasks.

Keywords: Diffusion model · Acceleration · Feature reuse

1 Introduction

The diffusion model has gained attention for its high-quality and diverse image generation capabilities [30,31,33,34]. Its outstanding quality and versatility unlocked new potentials across various applications, including image restoration [17,43], image editing [3,8,32,42,44], conditional image synthesis [1,7,27,45,47,49,50], and more. However, the substantial computation cost of the diffusion model, particularly due to its dozens to hundreds of denoising steps, poses a significant obstacle to its widespread adoption. To fully harness the benefits of diffusion models in practice, this performance drawback must be addressed.

Recently, many studies have proposed methods to mitigate the computational burden of diffusion models. A representative approach involves a zero-shot sampling method [18,39,41], which typically employs advanced ODE or SDE solvers

[*] Equal Contribution



Fig. 1: SDXL Acceleration with FRDiff

capable of maintaining quality with a reduced number of score function evaluations (NFE). While these methods demonstrate the potential for acceleration without fine-tuning, the performance improvement achievable within the accuracy margin is often insufficient. On the other hand, another direction employs learning-based sampling methods [19,26,35,40], applying fine-tuning to preserve generation quality with a reduced NFE. However, the requirement of fine-tuning, such as additional resources and a complex training pipeline, makes it challenging to use in practice. *To realize performance benefits in practice with minimal constraints, we need more advanced zero-shot methods with higher potential.*

In this work, we focus on an important but overlooked aspect of the diffusion models. Since they entail iterative denoising operations, **the feature maps within the diffusion models exhibit temporal redundancy**. According to our extensive analysis, specific modules within diffusion models show considerable similarity in their feature maps across adjacent frames. By reusing these intermediate feature maps with higher temporal similarity, we can significantly reduce computation overhead while maintaining output quality. Building on this insight, we propose a new optimization potential named **feature reuse (FR)**. However, the naive use of FR doesn't guarantee superior performance compared to the conventional reduced NFE method. Our thorough experiments reveal that FR has distinctive characteristics compared to reduced NFE methods, and both methods can complement each other to maximize the benefits we can achieve.

Overall, **we propose a comprehensive method named FRDiff, designed to harness the strengths of both the reduced NFE and FR**. Specifically, we introduce a score mixing technique to generate high-quality output with fine details. Additionally, we design a simple auto-tuning, named **Auto-FR**, to optimize the hyperparameters of FR to maximize the outcome quality within given constraints, such as latency. This approach can be applied to any diffusion model without the need for fine-tuning in existing frameworks with minimal modification. We conduct extensive experiments to validate the effectiveness of FRDiff on various tasks in a zero-shot manner. We can achieve up

to a **1.76x** acceleration without compromising output quality across a range of tasks, including a task-agnostic pretrained model for text-to-image generation, as well as task-specific fine-tuned models for super resolution and image inpainting. Code is available at `https://github.com/ECoLab-POSTECH/FRDiff`.

2 Related Works

2.1 Diffusion Models

The diffusion model, introduced in [38], defines the forward diffusion process by gradually adding Gaussian noise at each time step. Conversely, the reverse process generates a clean image from random noise by gradually removing noise from the data. In DDPM [10], the authors simplified the diffusion process using a noise prediction network $\epsilon_\theta(x_t, t)$ and reparameterized the complex ELBO loss [14] into a more straightforward noise matching loss. On a different note, [41] transforms the forward process of the diffusion model into a Stochastic Differential Equation (SDE). More recently, Classifier-Free Guidance (CFG) [11] has been introduced to guide the score toward a specific condition c. In the CFG sampling process, the score is represented as a linear combination of unconditional and conditional scores.

Since FRDiff is formulated based on the temporal redundancy inherent in the iterative diffusion process, it can be seamlessly integrated into all the previously mentioned methods, providing benefits irrespective of their specific details.

2.2 Diffusion Model Optimization

To accelerate the generation of diffusion models, many studies have concentrated on reducing NFE, which can be broadly categorized into two groups: zero-shot sampling [12,18,20,21,39,46,51], applying optimization to the pre-trained model, and learning-based sampling [19, 22, 26, 35], involving an additional fine-tuning.

Zero-shot sampling methods typically employ advanced Ordinary Differential Equation (ODE) solvers capable of maintaining generation quality even with a reduced NFE. For instance, DDIM [39] successfully reduced NFE by extending the original DDPM to a non-Markovian setting and eliminating the stochastic process. Furthermore, methods utilizing Pseudo Numerical methods [18], Second-order methods [12], and Semi-Linear structures [20, 21] have been proposed to achieve better performance. Learning-based sampling finetunes the model to perform effectively with reduced NFE. For example, Progressive Distillation [35] distills a student model to achieve the same performance with half the NFE. Recently, the consistency model [22, 40] successfully reduced NFE to 1-4 by predicting the trajectory of the ODE.

In addition, there are studies aimed at optimizing the backbone architecture of the diffusion model. These studies involve proposing new diffusion model structures [13, 33], as well as lightweighting the model's operations through techniques such as pruning [6], quantization [15, 37], and attention acceleration [2].

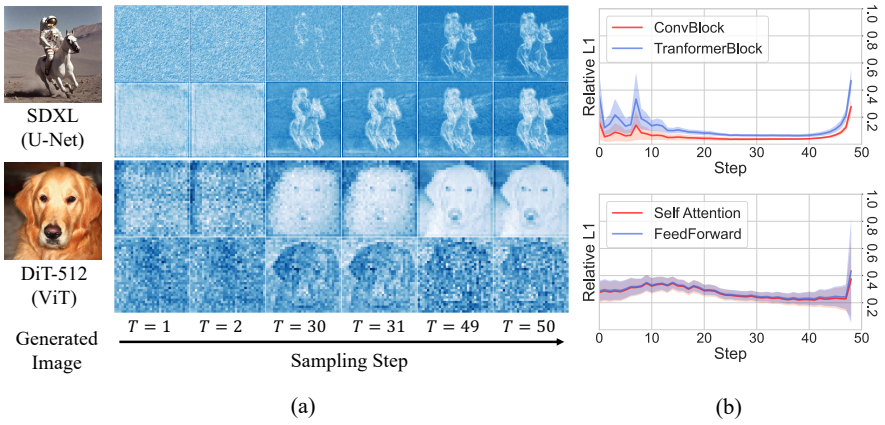


Fig. 2: Temporal Similarity Analysis of Diffusion Model: (a) Visualization of Intermediate Feature Maps During Inference. (b) Mean and Variance of Relative L1 Distance Between Adjacent Time Steps.

In this work, we primarily focus on enhancing the benefit of zero-shot model optimization for diffusion models. However, it's important to note that the proposed method could be applied in conjunction with other learning-based or backbone optimization studies.

3 Method

In this study, we introduce the idea of feature reuse (FR) as an innovative approach to expand the scope of model optimization for diffusion models. FR possesses distinct attributes compared to reduced NFE, enabling a synergistic effect when used together. In this chapter, we will explain the motivations behind the proposal of FR and discuss the expected advantages of this method.

3.1 Temporal Redundancy

Contemporary diffusion models often incorporate a series of blocks with a residual architecture. This design involves adding the layer's output to the input, as generally formulated by the following equation:

$$\mathbf{y}_i^t = \mathcal{F}_i(\mathbf{x}_i^t, t) + \mathbf{x}_i^t. \tag{1}$$

Here, i represents the index of layer, and $\mathcal{F}(\cdot)$ denotes the layer function, $\mathbf{x}, \mathbf{y}$ denote the input and output of the residual block, respectively, and $t \in [1, 2, ..., T]$ denotes the time step. Please note that Eq.1 incorporates temporal information as an input, allowing the model to be conditioned on time step. To generate high-quality images, the score estimation network $\epsilon(x_t, t)$ is repeatedly employed, taking the noisy image x_t at time step t as input and predicting the added noise.

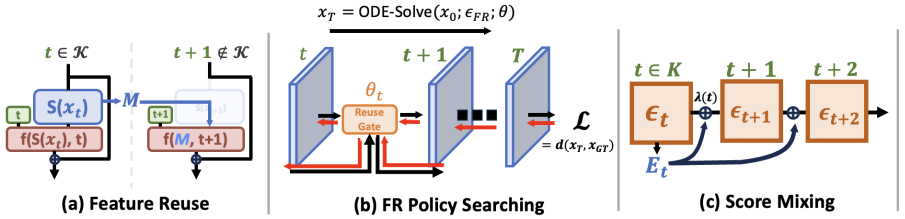


Fig. 3: Overview of our methods. (a) Feature Reuse (Eq. 5) (b) Auto-FR for Optimal $\mathcal{K}$ Searching (Eq. 13) (c) Score Mixing (Eq. 8)

The function $\mathcal{F}_i(\mathbf{x}_i^t, t)$ may take various forms depending on the architecture of diffusion model. For instance, in [33], $\mathcal{F}_i(\cdot)$ represents a convolutional layer or a self-attention layer with U-net structure, while in [29], it resembles a ViT-like Block. The detailed specification of $\mathcal{F}_i(\cdot)$ in various diffusion model is provided in the Appendix.

In this paper, our primarily focus on the temporal behavior of diffusion model, stemming from their repeated denoising operation. Specifically, we observe that the temporal changes of diffusion model remains relatively small across most time steps, regardless of the architecture and dataset. To aid the reader's comprehension, we offer a quantitative measure of *temporal change* in ith layer of diffusion model, as described by the following equation:

$$K_i(t, t') = \mathbb{E}_{\mathbf{x}}\left[\frac{\left\|\mathcal{F}_i(\mathbf{x}^t, t) - \mathcal{F}_i(\mathbf{x}^{t'}, t')\right\|_1}{\Delta t}\right], \tag{2}$$

where $\Delta t = \|t - t'\|$. In Fig 2(b), we showcase $K(t, t+1)$ across different layers of SDXL [30] and DiT [29]. As depicted, the temporal differences are minimal for the majority of time steps. We also provide visual evidence of this strong similarity in Fig 2(a). This similar appearance suggests that diffusion models may undergo redundant computations during the sampling process, indicating ample room for optimization.

3.2 Feature Reuse

Expanding the earlier observation, we introduce a method to remove the unnecessary computations in the denoising process. Specifically, we store the results of intermediate features from previous timesteps and reuse them in the subsequent timesteps. While this idea is simple and intuitive, we analyze its effects in depth and propose a novel approach to maximize the benefits we can derive.

First, we can decompose the computation of residual block into two parts:

$$\mathcal{F}_i(\mathbf{x}_i^t, t) = f_i(\mathcal{S}_i(\mathbf{x}_i^t), t), \tag{3}$$

where $\mathcal{S}(\cdot)$ is the operation performed before considering temporal information. Next, we define the *keyframe set* $\mathcal{K} \subseteq \{1, .., N\}$, which represents the set of

timesteps where the entire layers are updated and feature maps are saved for feature reusing. Here, N denotes the number of sampling steps for generation (e.g., $N = 50$ in DDPM).

For *keyframe* timestep $t \in \mathcal{K}$, The result of $\mathcal{S}(\cdot)$ is stored in memory $\mathbf{M}_i^t$ for future reuse. The remaining operation in residual block is performed normally. This process can be expressed as follows:

$$\begin{aligned} \mathbf{M}_i^t &\leftarrow \mathcal{S}_i(\mathbf{x}_i^t), \\ \mathbf{y}_i^t &= f_i(\mathbf{M}_i^t, t) + \mathbf{x}_i^t. \end{aligned} \tag{4}$$

For *non-keyframe* timesteps $t' \notin \mathcal{K}$, the computation of the $\mathcal{S}(\cdot)$ is replaced by the saved memory from the nearest early timestep $t \in \mathcal{K}$, as follows:

$$\mathbf{y}_i^{t'} = f_i(\mathbf{M}_i^t, t') + \mathbf{x}_i^{t'}. \tag{5}$$

Hence, by skipping the operations $\mathcal{S}(\cdot)$, a significant amount of computation can be saved, as illustrated in Fig. 3 (a). Moreover, since the sequential denoising operation of the diffusion model typically progresses from $t = 1$ to N, feature reuse can be implemented with a single memory M_i for each layer. An important point to note is that **the temporal information is updated while $\mathcal{S}(\cdot)$ is reused**. This allows the time-conditioned information to propagate through the residual path, enabling the diffusion model to be conditioned on the correct time step. Because this feature reuse scheme is highly flexible, it can be applied to any diffusion model architecture that utilizes skip connection, including both U-Net and diffusion transformer architectures.

3.3 Analysis: the effect of $\mathcal{K}$ selection

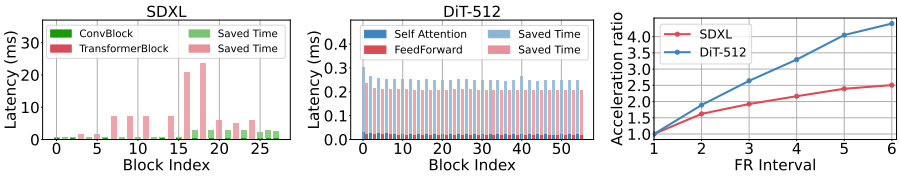
The generation quality and acceleration effect of FR can vary significantly depending on the appropriate keyframe set. In this section, we analyze the effect of keyframe set selection using heuristic design. The simplest way to construct a keyframe set is to compose it as a collection of timesteps with uniform intervals. This $\mathcal{K}_{uniform}$ is defined as follows:

$$\mathcal{K}_{uniform}^{M} = \{t \in \mathbb{N} \mid t \bmod M = 0, t \leq N\} \tag{6}$$

where **the FR interval** M represents how long the saved data will be reused.

Acceleration with Feature Reuse Because we only reuse a portion of the Residual block, we need to validate whether this method can indeed offer practical advantages. Therefore, we conducted a comprehensive analysis to evaluate the latency with FR on a real device (Gefore RTX 3090).

In Fig. 4, we depict the latency profiles of individual blocks within two different diffusion model architectures: U-Net(SDXL [30]) and DiT [29]. In the case of U-Net, the transformer block accounts for a larger portion of the execution cycle. This is primarily due to the spatial self-attention layer for high resolution



(a) SDXL [30] **(b)** DiT-512 [29] **(c)** Acceleration Ratio

Fig. 4: Detailed Analysis of Skippable Latency by Feature Reusing (FR) Technique for (a) SDXL (U-Net) and (b) DiT (Diffusion Transformer). (c) Speed-Up of FR Regarding FR Interval.

features. In the case of DiT, almost every block shows the same latency because it has consistent dimensions for every layer. For both architectures, we can save latency of approximately 92 percent and 68 percent, respectively, with FR.

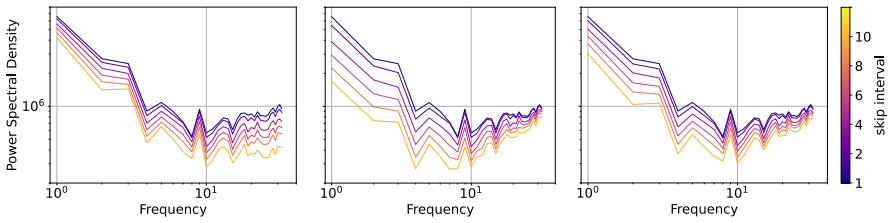
In Fig. 4(c), we measured the actual acceleration of FR against the FR interval. We use $N = 50$ in this experiment. As shown in the figure, because there are parts that are not skipped, the speed up saturates as the FR interval increases. However, even a small FR interval, e.g., 2 to 3, offers notable improvement.

Output Quality with Feature Reuse As explained in the previous section, FR could provide advantages in terms of both performance and quality. However, there are other acceleration methods such as the reduced NFE, so the adoption of FR should be justified by the distinctive advantages of FR over the reduced NFE. In this section, we will elucidate the unique benefits of FR we have discovered.

First, we explain the relation between the reduced NFE and FR. If we consider the very coarse-grained form of FR, which is reusing the entire output of the diffusion model $\epsilon(x, t)$, it is equivalent to the case of the reduced NFE. Indeed, FR is the fine-grained skipping in layer-wise granularity while the reduced NFE is the coarse-grained skipping in network-wise granularity. We provide a detailed interpretation of it for the DDIM case in the appendix.

Intuitively, because the reduced NFE skips more computation than FR, this should generate more degraded output. However, we observe interesting patterns in terms of *frequency response*. In Fig. 5, we depict the Power Spectral Density (PSD) analysis of generated images of the reduced NFE and FR. In the reduced NFE, the skip interval is increased from 1 to 10 while the FR interval ranges from 1 to 10 in FR. As shown in the figure, (a) the reduced NFE loses many high-frequency components while preserving the low-frequency area well. Meanwhile, (b) FR loses more low-frequency components while better preserving the high-frequency components. Please check the appendix for the visual comparison.

Our empirical findings suggest that FR is not consistently better than the reduced NFE; they possess distinct strengths and weaknesses. At this point, we need to pay attention to the recent findings on the generation characteristics. Recent studies [4, 6, 24, 48] have shown that in the early denoising stages, the model mainly generates coarse-grained low-frequency components. In contrast,



(a) Reduced NFE **(b)** Feature Reuse **(c)** Score Mixing

Fig. 5: Power Spectral Density (PSD) of Generated Image: (a) Reduced NFE, (b) Naive Feature Reuse, and (c) The Proposed Idea, Score Mixing

during the later stages, it predominantly generates fine-grained high-frequency components. Therefore, by combining these operational characteristics with our observations, **we devised a strategy that primarily employs the reduced NFE in the initial stages to generate coarse-grained structure and then switches to FR in the later stages to retain fine-grained details.**

3.4 Score Mixing

Unified view of FR and the reduced NFE : Score mixing To integrate the two methods, we introduce an additional heuristic called "score mixing". Instead of just switching from the reduced NFE to FR, we propose to use mixture of the output of this two method. Specifically, at $t \in \mathcal{K}$, we also save the output of the model $\epsilon(x_t, t)$ to memory,

$$E_t \leftarrow \epsilon(x_t, t) \tag{7}$$

Then, we employ a linear interpolation of the score estimated by FR and output from previous keyframe, controlled by mixing schedule $\lambda(t)$. This modified score is inputted into the next iteration step of diffusion model.

$$\epsilon(x_{t'}, t') \leftarrow \lambda(t') * \epsilon(x_{t'}, t') + (1 - \lambda(t')) * E_t \tag{8}$$

$$\lambda(t) = \max(0, \min(1, (\tau * ((t/N) - b) + 2)/4)). \tag{9}$$

While any increasing function can be used for $\lambda(t)$, we use the *hard sigmoid* function. By using this $\lambda(t)$, we can skip the computation of the FR score when $\lambda(t) = 0$. In the case of conditional sampling (CFG), we simply mix the conditional score in the same way as the unconditional score, using the same λ.

In Eq. 9, τ is the temperature that controls the switching speed of the schedule, and b is the bias that controls the phase transition point of the schedule. We empirically determine the optimal values as $\tau = 30$ and $b = 0.5$ and use these values throughout the rest of the paper. In Fig.5.(c), we depict the PSD analysis of the generated image using our score mixing. As shown in the figure, this exhibits the preservation of low and high frequencies compared to FR and the reduced NFE.

3.5 Auto-tuning for FR interval: Auto-FR

With score mixing, FR is used when $\lambda(t) > 0$. To maximize the benefit of FR, we propose an automated search called **Auto-FR** to find the optimal $\mathcal{K}$. In this approach, we apply a timestep-wise learnable parameter $\alpha_t = sigmoid(\theta_t)$ with a hard gating mechanism update. It's important to note that the network parameters are frozen and training-free; only the gating parameters are updated. The forward path of the residual block is computed as follows:

$$\alpha_t^* = \lfloor \alpha_t \rceil + \alpha_t - \texttt{stopgrad}(\alpha_t), \tag{10}$$

$$\mathbf{M}_i \leftarrow \alpha_t^* \cdot \mathcal{S}_i(\mathbf{x}_i^t) + (1 - \alpha_t^*) \cdot \mathbf{M}_i. \quad \mathbf{y}_i^t = f_i(\mathbf{M}_i, t) + \mathbf{x}_i^t \tag{11}$$

Likewise, the forward path of score mixing is computed with gate parameter :

$$\mathbf{E} \leftarrow \alpha_t^* \cdot \epsilon(x_t, t) + (1 - \alpha_t^*) \cdot \mathbf{E}. \quad \epsilon(x_t, t) \quad = \lambda(t) * \epsilon(x_t, t) + (1 - \lambda(t)) * \mathbf{E} \tag{12}$$

With this scheme, we can safely simulate and differentiate through the sampling process of FR if $t = 0 \rightarrow N$. We update θ_t using gradient descent with straight-through estimation to minimize the following loss function $\mathcal{L}(\cdot)$:

$$\mathcal{L}(\theta) = \mathbb{E}_x \left[\left\| x^{GT} - \texttt{ODE-Solve}(x_0; N, \theta) \right\|_2 \right] + \lambda * \mathcal{L}_{cost}(\theta) \tag{13}$$

Here, x^{GT} is the ground-truth sample generated from noise x_0, `ODE-Solve` is a differentiable ODE Solver (e.g., DDIM), $\mathcal{L}_{cost}$ is the latency cost of FR, and λ is the balancing parameter that effectively controls the trade-off between latency and fidelity. The detailed training recipe is provided in the appendix. In short, the reuse policy is trained to maximize quality while minimizing the computation cost of the sampling process.

Finally, the keyframe set is determined from the trained θ^*:

$$\mathcal{K}_{search}^{\lambda} = \{t \in \mathbb{N} \mid \theta_t^* \geq 0, 0 \leq t \leq N\}. \tag{14}$$

Our final solution, FRDiff, is the mixture of score mixing and Auto-FR, designed to leverage the benefits of the reduced NFE and FR with minimal human intervention.

4 Experiments

4.1 Experiments Setup

To validate the effectiveness of our proposed idea, we assessed its efficacy across various existing diffusion models, including pixel-space (CIFAR-10) [10], latent diffusion model (LDM) [33], and Diffusion Transformer (DiT) [29]. Our intentional choice of diverse models aimed to demonstrate the versatility of the proposed idea. For example, the pixel-space model and LDM utilize a U-net structure, while DiT employs the diffusion transformer instead of U-net. Our method

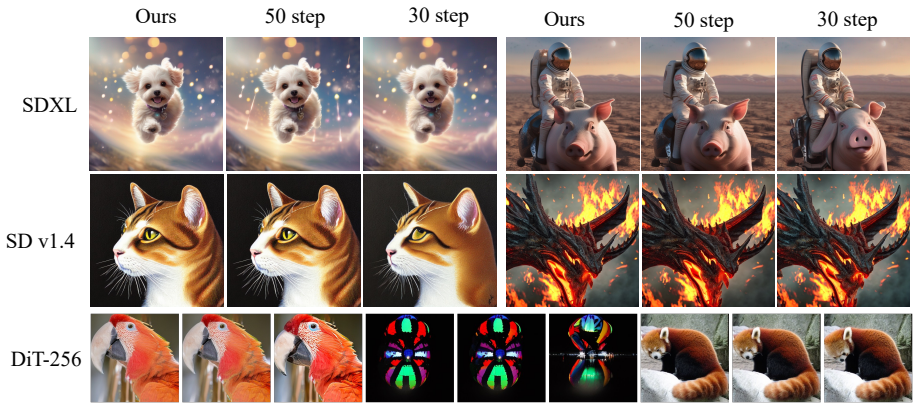


Fig. 6: Qualitative Comparison of FRDiff. *Best viewed zoomed-in.* (left) Ours (middle) DDIM 50 step baseline (right) DDIM 30 step. Our FRDiff up to 1.76x faster than the baseline, DDIM 50.

Table 1: Quantitative Results of FRDiff.

Model	NFE	FRDiff	Latency(s)↓	Speed-up↑	FID↓
CIFAR-10 [10]	50		0.836	1.00x	4.03
	30		0.495	1.68x	5.01
	50	✓	**0.491**	**1.70x**	**4.64**
LDM-CelebA [33]	50		1.317	1.00x	6.0
	30		0.763	1.72x	7.22
	50	✓	**0.748**	**1.76x**	**6.33**
SD v1.4 [33]	30		4.657	1.00x	6.32
	20		3.027	1.53x	8.41
	30	✓	**2.947**	**1.58x**	**6.86**
SDXL [30]	30		8.810	1.00x	7.41
	20		6.033	1.46x	9.46
	30	✓	**5.491**	**1.60x**	**9.28**
DiT-256 [29]	30		0.763	1.00x	14.76
	20		0.515	1.48x	17.69
	30	✓	**0.463**	**1.64x**	**16.71**

is designed to be applicable across all these model architectures. We obtained pretrained weights from the official repository for all models except for the pixel-space model (CIFAR-10).

For evaluation, we conducted both qualitative and quantitative experiments. In the qualitative assessment, we compared our generated images against the baseline images produced by DDIM with 50 steps [39] in terms of fidelity and latency, as shown in Fig. 6. For quantitative evaluation, we measured the Fréchet

Inception Distance (FID) [9] for Stable Diffusion(SD) [33] and SDXL [30] using 5k samples in MS-COCO [16] and for DiT using 10k samples in ImageNet [5]. We also measure the speedup compared to the DDIM baseline, as summarized in Table 1. All experiments were conducted on a GPU server equipped with an NVIDIA GeForce RTX 3090, and latency measurements were performed using PyTorch [28] with a batch size of 1 on a GeForce RTX 3090.

Qualitative Analysis In Fig 6, we present a comparison of image generation results using various diffusion models: SDXL [30], Stable Diffusion (SD) [33], and DiT-256 [29]. For comparison, we depict the baseline image (middle; DDIM with 50 steps), Our method (left; baseline + FRDiff), and DDIM with 30 steps, as fast as ours (right). As shown in the figure, Our method (left) can accelerate the baseline (middle) without quality degradation, while DDIM with reduced NFE (right) shows severe quality degradation, exhibiting notable artifacts. Our method can safely accelerate the diffusion sampling process up to 1.76x (average 1.62x) regardless of architecture and dataset.

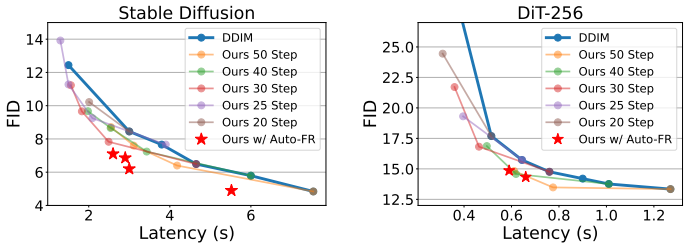
Quantitative Analysis In table 1, we also measured the FID score and relative speedup of baseline(DDIM 50), Ours(DDIM 50 + FRDiff), and the reduced NFE as fast as ours (DDIM 30) in various diffusion models. As shown in table, ours demonstrate superior performance than DDIM in terms of both latency and fidelity, regardless of dataset and architecture. For a more detailed analysis, please refer to the Pareto-line analysis, presented in Sec. 4.3 and Fig. 7.

4.2 Comparison with Existing Methods

In Table 2, we compare our method with several recently developed fast sampling methods for diffusion models. Specifically, we categorize these methods into three types: distillation-based methods [23, 36], advanced ODE solvers [39, 52], and other training-free acceleration methods [25], which enable zero-shot acceleration. For distillation-based methods, although they achieve extremely low latency, their FID scores are relatively large, limiting their utility in cases where image quality is paramount. Additionally, the substantial training costs associated with these methods pose a significant barrier to widespread adoption. Next, we compare our method with recently proposed fast ODE solvers such as DDIM [39] and DPM-Solver++ [52]. Our method demonstrates a smaller quality degradation compared to these recent ODE solvers, leveraging the unexplored potential of diffusion acceleration enabled by FR. Finally, we compare our method with DeepCache [25], which also aims to accelerate the denoising process through feature caching. DeepCache saves intermediate activations in a depth-wise, coarse-grained manner. However, our FRDiff exhibits a better latency-tradeoff than DeepCache due to its fine-grained feature utilization and judicious design with score mixing and Auto-FR. Moreover, it's worth noting that while DeepCache's feature reusing scheme relies on UNet architecture, our method is applicable to any architecture that has a residual structure.

Table 2: Comparison of Existing Diffusion Acceleration Methods.

Method	NFE	Latency↓	Retrain	FID↓
SDXL-Turbo [36]	4	0.731	✓	22.58
LCM [23]	1	0.171	✓	42.53
DDIM [39]	30	4.654	✗	6.31
	20	3.078	✗	8.45
DPM-Solver++ [52]	20	3.075	✗	6.63
DeepCache [25]	50	5.027	✗	6.34
Ours (M=2)	50	4.183	✗	6.40
Ours (M=3)	50	3.117	✗	7.60
Ours (M=2)	40	3.333	✗	7.24
Ours (M=2)	30	2.492	✗	7.83
Ours w/ AutoFR	35	2.914	**0.1h**	**6.20**



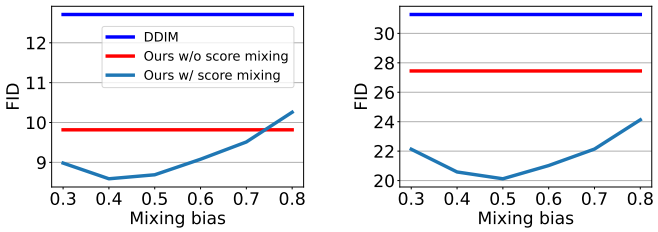
(a) Stable-Diffusion **(b)** DiT-256

Fig. 7: Pareto-line Comparison of the reduced NFE vs FRDiff

4.3 Ablation Study

reducing NFE vs FR In this section, we visualize the effect of changing the NFE and FR Interval (M) on SDXL [30] in Fig. 9. As depicted, reducing the NFE leads to a rapid degradation in performance, whereas reducing the Keyframe ratio maintains performance relatively well. Although FR incurs slightly more computation than reduced NFE, it offers new opportunities in the quality-latency trade-off.

To explore the trade-off relationship in detail, we draw Pareto lines in terms of latency and FID. In this experiment, the blue solid line represents the DDIM results with only reduced NFE. Meanwhile, our method adjusts both NFE and FR interval simultaneously; each line represents the corresponding reduced NFE, while the data point is added by increasing the FR interval gradually. As shown in the figure, adjusting both the keyframe ratio and NFE (Ours) clearly shows better Pareto fronts than DDIM. Furthermore, in Fig. 7, we depict the points



(a) Skip interval 4 **(b)** Skip interval 8

Fig. 8: Impact of Bias Value on FID in Score Mixing.

Table 3: Memory Overhead

Model	Size (MB)
CIFAR-10	5.4
LDM-4	44.4
SD v1.4	180.0
SDXL	530.0
DiT-512	252.0

obtained by Auto-FR as $\star$ with different cost objectives. These points exhibit superior latency-FID Pareto fronts, illustrating the benefits of autoML-based optimization. We provide a detailed training recipe and trained keyframe sets of these points in the appendix. Our Auto-FR can find an optimal FR policy with minimal training costs, enabling us to exploit the benefits of FR with minimal human intervention.

Score Mixing In Eq. 9, the bias b determines the transition point from the reduced NFE for low-frequency components to FR for high-frequency ones. To understand the impact of this bias on generation performance, we swept the bias from 0.3 to 0.8 and measured the FID score, as shown in Fig. 8. The figure illustrates that the optimal FID is achieved when the bias is around 0.4 to 0.5, indicating that the mixture of the reduced NFE and FR is definitely helpful in improving accuracy, with the advantageous interval being approximately half and half. Based on this observation, we empirically set $b = 0.5$ throughout the rest of the paper.

Memory Overhead FR needs to save the intermediate features for future timesteps, necessitating additional memory space. Table 3 presents the total amount of memory required for feature reusing. As shown in the table, our method can be applicable within an affordable memory overhead. For instance, in DiT-512, the model takes 4.2 GB of running memory; only adding 5.8% of additional space, we enjoy 1.64x faster generation speed.

Other Tasks To assess the versatility of our approach, we apply it to several other tasks including super-resolution, image inpainting, and text-to-video generation. FRDiff proved to be applicable across all tasks, safely accelerating existing denoising processes. Additional results are provided in the Appendix.

5 Discussion

In this section, we will address the limitations of our approach. Although our method offers the advantage of plug-and-play integration without reliance on

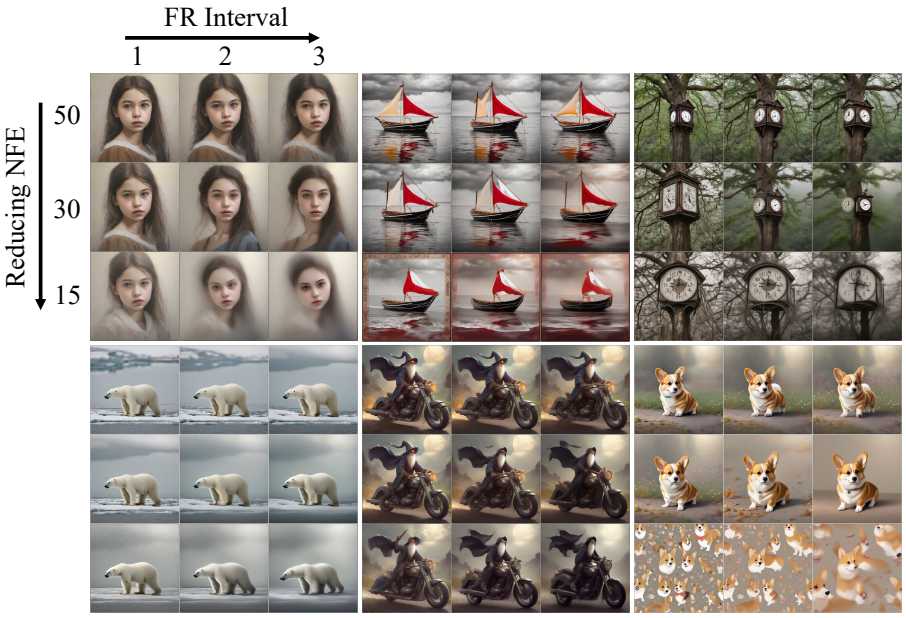


Fig. 9: Visualization of Reduced NFE vs FR. *Best viewed zoomed-in.*

a particular ODE solver, its applicability becomes non-trivial when the time step for score function evaluation is not continuously provided. For example, in DPM-Solver++ [21], where 2 or 3 non-consecutive score function evaluations are needed to compute a single score, the efficacy of FR may diminish due to non-consecutive feature maps. Further investigation is warranted for such methods.

6 Conclusion

In this paper, we introduce *FRDiff*, a novel Feature Reusing (FR)-based zero-shot acceleration technique for diffusion models. By leveraging the temporal similarity inherent in the iterative generation process, FRDiff can achieve remarkable acceleration of up to 1.76x without sacrificing output quality. Moreover, through a comprehensive examination, we present two additional techniques: score mixing, which harnesses the advantages of both reduced NFE and FR, and Auto-FR, which determines the optimal configuration through automated tuning. We validate our approach across diverse task datasets, demonstrating superior generation quality compared to existing acceleration methods within the same latency constraints.

Acknowledgments This work was supported by IITP grant funded by the Korea government (RS-2023-00228970, RS-2021-II210310, RS-2021-II210105, and RS-2019-II191906) and Samsung Research Global AI Center.

References

1. Bansal, A., Chu, H.M., Schwarzschild, A., Sengupta, S., Goldblum, M., Geiping, J., Goldstein, T.: Universal guidance for diffusion models. In: CVPR. pp. 843–852 (2023)
2. Bolya, D., Hoffman, J.: Token merging for fast stable diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4598–4602 (2023)
3. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: CVPR. pp. 18392–18402 (2023)
4. Choi, J., Lee, J., Shin, C., Kim, S., Kim, H., Yoon, S.: Perception prioritized training of diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11472–11481 (2022)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
6. Fang, G., Ma, X., Wang, X.: Structural pruning for diffusion models. arXiv preprint arXiv:2305.10924 (2023)
7. Ham, C., Hays, J., Lu, J., Singh, K.K., Zhang, Z., Hinz, T.: Modulating pretrained diffusion models for multimodal image synthesis. SIGGRAPH Conference Proceedings (2023)
8. Hertz, A., Aberman, K., Cohen-Or, D.: Delta denoising score. In: ICCV. pp. 2328–2337 (2023)
9. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., 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)
10. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
11. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
12. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. Advances in Neural Information Processing Systems **35**, 26565–26577 (2022)
13. Kim, B.K., Song, H.K., Castells, T., Choi, S.: On architectural compression of text-to-image diffusion models. arXiv preprint arXiv:2305.15798 (2023)
14. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
15. Li, X., Liu, Y., Lian, L., Yang, H., Dong, Z., Kang, D., Zhang, S., Keutzer, K.: Q-diffusion: Quantizing diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17535–17545 (2023)
16. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., 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)
17. Lin, X., He, J., Chen, Z., Lyu, Z., Fei, B., Dai, B., Ouyang, W., Qiao, Y., Dong, C.: Diffbir: Towards blind image restoration with generative diffusion prior. arXiv preprint arXiv:2308.15070 (2023)
18. Liu, L., Ren, Y., Lin, Z., Zhao, Z.: Pseudo numerical methods for diffusion models on manifolds. arXiv preprint arXiv:2202.09778 (2022)

19. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. arXiv preprint arXiv:2209.03003 (2022)
20. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. Advances in Neural Information Processing Systems **35**, 5775–5787 (2022)
21. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. arXiv preprint arXiv:2211.01095 (2022)
22. Luo, S., Tan, Y., Huang, L., Li, J., Zhao, H.: Latent consistency models: Synthesizing high-resolution images with few-step inference. arXiv preprint arXiv:2310.04378 (2023)
23. Luo, S., Tan, Y., Huang, L., Li, J., Zhao, H.: Latent consistency models: Synthesizing high-resolution images with few-step inference. arXiv preprint arXiv:2310.04378 (2023)
24. Ma, H., Zhang, L., Zhu, X., Feng, J.: Accelerating score-based generative models with preconditioned diffusion sampling. In: European Conference on Computer Vision. pp. 1–16. Springer (2022)
25. Ma, X., Fang, G., Wang, X.: Deepcache: Accelerating diffusion models for free. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15762–15772 (2024)
26. Meng, C., Rombach, R., Gao, R., Kingma, D., Ermon, S., Ho, J., Salimans, T.: On distillation of guided diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14297–14306 (2023)
27. Mou, C., Wang, X., Xie, L., Zhang, J., Qi, Z., Shan, Y., Qie, X.: T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. arXiv preprint arXiv:2302.08453 (2023)
28. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
29. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4195–4205 (2023)
30. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952 (2023)
31. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: International Conference on Machine Learning. pp. 8821–8831. PMLR (2021)
32. Ravi, H., Kelkar, S., Harikumar, M., Kale, A.: Preditor: Text guided image editing with diffusion prior. arXiv preprint arXiv:2302.07979 (2023)
33. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
34. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. NeurIPS **35**, 36479–36494 (2022)
35. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. arXiv preprint arXiv:2202.00512 (2022)
36. Sauer, A., Lorenz, D., Blattmann, A., Rombach, R.: Adversarial diffusion distillation. arXiv preprint arXiv:2311.17042 (2023)

37. So, J., Lee, J., Ahn, D., Kim, H., Park, E.: Temporal dynamic quantization for diffusion models. arXiv preprint arXiv:2306.02316 (2023)
38. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning. pp. 2256–2265. PMLR (2015)
39. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
40. Song, Y., Dhariwal, P., Chen, M., Sutskever, I.: Consistency models. arXiv preprint arXiv:2303.01469 (2023)
41. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
42. Tumanyan, N., Geyer, M., Bagon, S., Dekel, T.: Plug-and-play diffusion features for text-driven image-to-image translation. In: CVPR. pp. 1921–1930 (2023)
43. Wang, J., Yue, Z., Zhou, S., Chan, K.C., Loy, C.C.: Exploiting diffusion prior for real-world image super-resolution. arXiv preprint arXiv:2305.07015 (2023)
44. Wang, Q., Zhang, B., Birsak, M., Wonka, P.: Mdp: A generalized framework for text-guided image editing by manipulating the diffusion path. arXiv preprint arXiv:2303.16765 (2023)
45. Xie, J., Li, Y., Huang, Y., Liu, H., Zhang, W., Zheng, Y., Shou, M.Z.: Boxdiff: Text-to-image synthesis with training-free box-constrained diffusion. In: ICCV. pp. 7452–7461 (2023)
46. Xu, Y., Deng, M., Cheng, X., Tian, Y., Liu, Z., Jaakkola, T.: Restart sampling for improving generative processes. arXiv preprint arXiv:2306.14878 (2023)
47. Yang, B., Luo, Y., Chen, Z., Wang, G., Liang, X., Lin, L.: Law-diffusion: Complex scene generation by diffusion with layouts. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22669–22679 (2023)
48. Yang, X., Zhou, D., Feng, J., Wang, X.: Diffusion probabilistic model made slim. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22552–22562 (2023)
49. Yu, J., Wang, Y., Zhao, C., Ghanem, B., Zhang, J.: Freedom: Training-free energy-guided conditional diffusion model. ICCV (2023)
50. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: CVPR. pp. 3836–3847 (2023)
51. Zhang, Q., Tao, M., Chen, Y.: gddim: Generalized denoising diffusion implicit models. arXiv preprint arXiv:2206.05564 (2022)
52. Zheng, K., Lu, C., Chen, J., Zhu, J.: Dpm-solver-v3: Improved diffusion ode solver with empirical model statistics. arXiv preprint arXiv:2310.13268 (2023)