

Rethinking Token Reduction for Diffusion Models via Output-Similarity-Awareness

Hangyeol Lee[✉], Hyojeong Lee[✉], and Joo-Young Kim[✉]

KAIST, Daejeon, Republic of Korea
{lhg4294,hjlee8877,jooyoung1203}@kaist.ac.kr

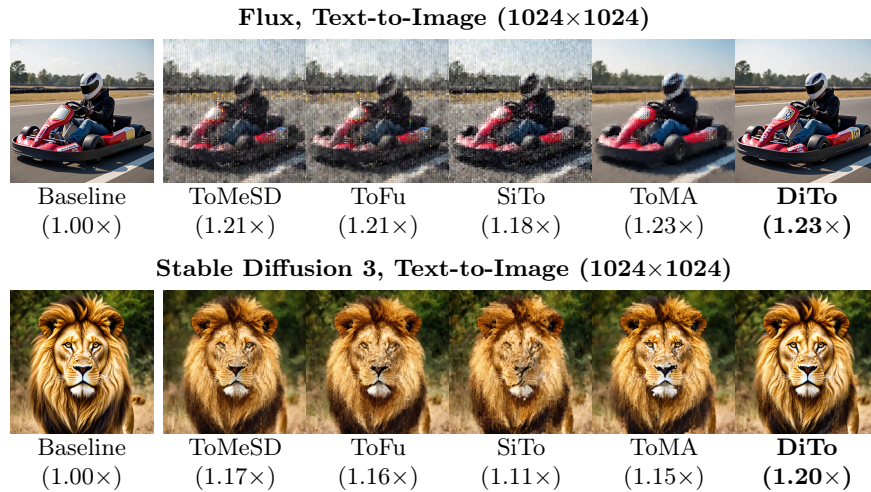


Fig. 1: Visual comparisons on Flux and SD3 with prior token reduction methods. Under the equivalent FLOPs reduction, DiTo preserves superior visual fidelity and structural consistency with comparable latency, while others suffer from severe structural distortion.

Abstract. Diffusion Transformers (DiTs) achieve superior image generation quality but suffer from quadratic computational complexity relative to token count. While various token reduction (TR) methods have been proposed to mitigate this cost, they overlook the primary objective of generative models: minimizing recovery error, which requires reflecting output token similarity. Specifically, they rely solely on input token similarity inherited from reduction-only ViT paradigms, leading to a fundamental misalignment with this objective.

To bridge this gap, we propose **DiTo**, a novel TR paradigm that shifts the focus toward output-centric token reduction. Based on the observation that output token similarity is consistently preserved across adjacent timesteps, DiTo utilizes prior-step similarities as an effective proxy to establish token correspondences at a Matching timestep, and reuses the

resulting correspondences across multiple subsequent Reduction steps. To optimize this interleaved scheduling, we propose Pair Match Ratio (PMR)-guided Interval Scheduling to determine the optimal matching frequency. Furthermore, to mitigate localized approximation errors and resulting blocking artifacts caused by repeated reuse, we propose Frequency-aware Token Matching by incorporating a selection-frequency penalty. Extensive experiments demonstrate that DiTo consistently outperforms existing TR methods with 1.6–3.9 dB higher PSNR at comparable speedups, achieving a superior Pareto frontier.

Keywords: Diffusion Model · Token Reduction Method

1 Introduction

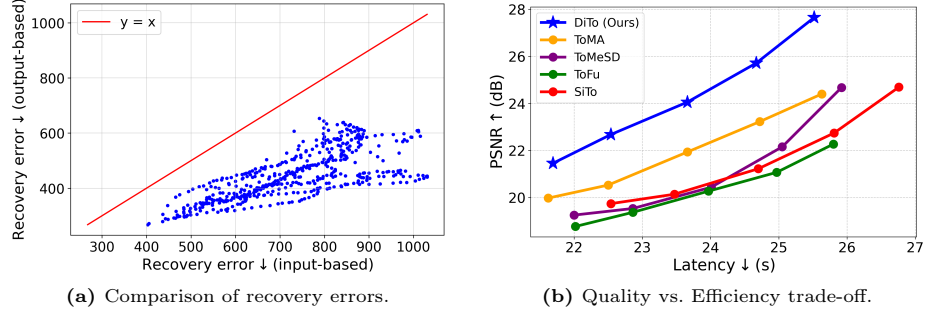


Fig. 2: Analysis of recovery error and performance trade-offs. (a) All 500 randomly selected samples fall below the $y = x$ line, indicating that output-based TR methods consistently achieve lower recovery error than existing input-based methods. (b) DiTo achieves a superior Pareto frontier in terms of generation quality (PSNR) and efficiency compared to existing methods.

Diffusion Transformers (DiTs) [3,6,15,30] have emerged as a powerful paradigm for high-fidelity image synthesis, primarily due to their scalable architecture composed of transformer blocks. Despite their impressive performance, the attention mechanism—the core component of transformers—exhibits quadratic complexity relative to the number of tokens. This scaling property leads to prohibitive computational overhead, establishing a significant bottleneck, particularly for high-resolution generation where the token count is substantial.

While FlashAttention [5] and xFormers [17] have successfully accelerated transformers through kernel-level optimization, they do not address the underlying computational complexity. To mitigate this, token reduction (TR) approaches have emerged as a promising solution [1,7,27,41], aiming to directly reduce the token count by pruning or merging redundant tokens. For Vision Transformers (ViTs), these methods are primarily designed for discriminative

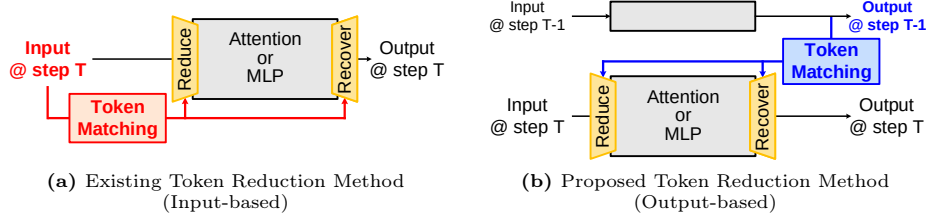


Fig. 3: Comparison of token reduction methods. (a) Existing input-based methods rely on current input token similarity. (b) The output-based method utilizes prior output token similarity, minimizing recovery error and enhancing generation quality.

tasks, focusing only on redundancy removal [1, 7, 13, 14, 19, 25, 27, 29, 32, 33, 40, 41]. In contrast, generative tasks like Diffusion Models (DMs) necessitate an additional recovery stage, making such reduction-only methods difficult to apply directly.

In DMs, tokens correspond to specific spatial positions within the output; thus, the original token count must be preserved through an explicit recovery stage. Accordingly, the TR process begins with the token matching stage, which establishes token correspondences that govern both the reduction and recovery stages by specifying how tokens are to be reduced and subsequently restored. Existing TR methods [2, 13, 23, 37, 44] inherit ViT-style paradigms that rely on input token similarity for the matching stage. However, within this recovery-inclusive TR framework, the primary objective shifts toward minimizing the recovery error—the discrepancy between the restored output and the original dense output. Since the recovery stage reconstructs reduced tokens by copying the outputs of their matched tokens, the error is fundamentally determined by how closely the matched tokens’ outputs represent the original outputs of the reduced ones. Therefore, to ensure a high-fidelity reconstruction, TR must be designed to reflect output token similarity rather than relying on input token similarity, as existing TR methods do. Our empirical analysis in Fig. 2a supports this necessity by evaluating the recovery errors. In this scatter plot, the x and y axes represent the recovery errors when applying TR via input-based and output-based matching, respectively. Notably, every evaluated sample falls below the $y = x$ line, demonstrating that output-based TR consistently achieves lower recovery errors than conventional input-based TR, which fail to align with the goal of error minimization.

To bridge this gap, we propose **Diffusion-specific Token Reduction (DiTo)**, which marks a paradigm shift towards output-centric token reduction, explicitly reflecting output token similarity to minimize recovery error. Since the current output is not yet available during the token matching stage, DiTo exploits the temporal consistency inherent in DMs as an effective proxy. While it is well-established that activations remain highly correlated across adjacent timesteps [4, 43, 47], we further observe that token similarity is likewise consistently preserved. Based on this insight, we leverage the output token similarity

from the prior timestep to guide the matching stage for the current reduction-recovery stages (Fig. 3b). This strategy partitions the diffusion timesteps into an interleaved schedule of matching timesteps, where token correspondences are established, and subsequent reduction timesteps that reuse these correspondences across multiple steps to perform efficient reduction and recovery. To effectively optimize the matching intervals, we introduce the Pair Match Ratio (PMR), a new metric that quantifies the temporal alignment with the current-step output’s matching results. Since DiTo relies on prior-step similarities, PMR serves as a reliable indicator for determining how long a matching result remains valid before alignment degrades. Leveraging this metric, we propose PMR-guided Interval Scheduling to find an optimal balance between computation efficiency and matching accuracy in an offline manner. While DiTo improves efficiency through the reuse of matching results, repeated application can exacerbate token-selection imbalance for reduction, concentrating approximation errors in localized regions and causing blocking artifacts. To mitigate this, we propose Frequency-aware Token Matching, which incorporates a similarity penalty based on token selection history to effectively suppress these artifacts.

With these designs, DiTo consistently outperforms existing input-based token reduction methods on representative DiT models, including Flux and SD3. DiTo markedly outperforms existing methods in visual preservation (Fig. 1), achieving 1.6–3.9 dB higher PSNR at similar or slightly higher speedups. These results demonstrate that DiTo effectively breaks the quality-efficiency trade-off, achieving a superior Pareto frontier (Fig. 2b).

2 Related Work

2.1 Acceleration for DMs

While diffusion models deliver exceptional generative performance, they incur high computational costs due to the extensive number of sampling timesteps and the heavy computational cost required per step. Accordingly, acceleration methods primarily focus on either reducing the sampling steps T or lowering the per-step cost. To reduce T , sampler-based approaches [21, 22, 38, 46, 48] employ higher-order numerical solvers, while distillation-based approaches [24, 26, 36, 42] compress multi-step teacher models into few-step students. But these approaches typically require additional training.

To reduce the per-step cost, various methods have been proposed: layer-distillation methods [12, 16] reduce the number of layers by training a shallower student model, pruning methods [8, 49] lower model complexity through parameter removal, and quantization methods [9, 18, 20] lower numerical precision to reduce memory and computational cost. Orthogonal to the aforementioned approaches, token reduction methods [2, 13, 23, 44] directly reduce the token length, effectively mitigating the quadratic attention cost. However, existing methods do not explicitly minimize recovery error, which can lead to noticeable quality degradation.

2.2 Token Reduction in ViTs

Token reduction (TR) has been extensively explored to enhance the efficiency of Vision Transformers (ViTs). The core objective of these methods is to identify redundant tokens based on token similarity or importance scores, thereby reducing the token count via pruning or merging. Importance-based methods [7, 14, 19, 27, 29, 33, 41] reduce uninformative tokens using importance scores derived from a trainable prediction module or the attention maps in self-attention layers. In contrast, Similarity-based methods [1, 13, 25, 32, 40] aggregate spatially redundant tokens by leveraging their token similarity. However, these methods are primarily tailored for discriminative tasks, which focus solely on redundancy removal. In contrast, since generative models such as Diffusion Models (DMs) require an additional recovery stage, these reduction-only methods cannot be directly applied.

2.3 Token Reduction in DMs

In DMs, tokens correspond to specific spatial positions within the final output; thus, the original token count must be preserved through an explicit recovery stage. Recently, several TR methods incorporating such recovery stages have been proposed for DMs [2, 13, 23, 44]. These approaches generally leverage input token similarity to identify redundant tokens and their corresponding reference tokens, forming matched token pairs to apply merging or pruning. For instance, ToMeSD [2] utilizes a bipartite soft matching algorithm to efficiently find and merge matched pairs, while ToFu [13] achieves higher performance by hybridizing merging and pruning strategies. Furthermore, SiTo [44] calculates per-token similarity scores based on global token relationships to select effective reference tokens, and ToMA [23] identifies optimal token pairs through a parallel greedy search within local regions.

However, these methods primarily focus on optimizing the token pair selection process by relying on input token similarity—a practice inherited from ViTs—which is sub-optimal from the perspective of minimizing recovery error. In contrast, DiTo explicitly minimizes recovery error by leveraging output token similarity to guide the TR process, thereby achieving superior generation quality compared to existing methods.

3 Preliminaries

3.1 Diffusion Transformers

Self-attention is the core computation in Diffusion Transformers (DiTs). For clarity, we present the essential operations of self-attention as follows:

$$\text{Attn}(X) = \text{softmax}(QK^\top) V = \text{softmax}((XW_Q)(XW_K)^\top)(XW_V), \quad (1)$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$ represent the projection matrices that produce the query, key, and value, respectively, and $X \in \mathbb{R}^{N \times d}$ denotes the input tensor of

attention. N represents the token length and d represents the hidden dimension. As shown in Eq. (1), the computational complexity scales quadratically with the token length, i.e., $O(N^2)$, leading to a rapid increase in computation and making it a major bottleneck in Transformers. Accordingly, token reduction (TR) methods effectively accelerate DiTs by directly reducing the token length, thereby alleviating this bottleneck and lowering the overall computational cost.

3.2 Token Reduction Pipeline: Formulation and Objectives

Formulation. We summarize the TR process in DMs through a unified pipeline that generalizes existing TR methods. The pipeline primarily consists of three distinct stages: token matching, reduction, and recovery. First, the token matching stage establishes token correspondences based on their similarity, providing the essential mapping information required for the subsequent reduction and recovery stages. This stage begins with token bipartitioning, where the total N tokens are split into two disjoint sets: a destination (dst) set $\mathcal{D}$ and a source (src) set $\mathcal{S}$, such that $N = |\mathcal{D}| + |\mathcal{S}|$. Subsequently, we perform dst–src pair matching by constructing a similarity map $\mathbf{A}$ between these two sets. For each src token $s \in \mathcal{S}$, we determine the most similar dst token and its corresponding similarity score as follows:

$$d^*(s) = \arg \max_{d \in \mathcal{D}} \mathbf{A}_{d,s}, \quad \hat{s}(s) = \max_{d \in \mathcal{D}} \mathbf{A}_{d,s}, \quad (2)$$

where $d^*(s)$ forms candidate token pairs and $\hat{s}(s)$ represents the candidate similarity scores. To determine the final matched token pairs, we select a subset of source tokens, $\mathcal{S}_K \subset \mathcal{S}$, consisting of the top- k tokens with the largest candidate similarity scores, where k is determined by a predefined reduction ratio. This results in the final set of matched pairs, $\mathcal{P} = \{(d^*(s), s) \mid s \in \mathcal{S}_K\}$. Finally, the source tokens $s \in \mathcal{S}_K$ defined in $\mathcal{P}$ are reduced to decrease the token count for efficient computation during the reduction stage. Following the core operations, these reduced tokens are restored during the recovery stage by copying the output of their corresponding $d^*(s)$ as specified in $\mathcal{P}$, ultimately yielding the restored output $\tilde{\mathbf{Y}}$.

Objectives. Unlike reduction-only TR in ViTs, DMs must include a recovery stage to restore the original token count. While solely identifying redundant tokens is sufficient in ViTs, DMs require this restoration to achieve high-fidelity reconstruction. Consequently, the primary objective in DMs is to ensure that the restored output $\tilde{\mathbf{Y}}$ closely approximates the dense output $\mathbf{Y}$ computed without TR. This goal is formally expressed as minimizing the recovery error ($\mathcal{L}_{\text{out}}$):

$$\min \mathcal{L}_{\text{out}} = \left\| \mathbf{Y} - \tilde{\mathbf{Y}} \right\|_F^2, \quad (3)$$

where $\|\cdot\|_F^2$ denotes the squared Frobenius norm between $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$.

To minimize the recovery error $\mathcal{L}_{\text{out}}$, it is essential to focus on the underlying mechanism of the recovery stage. Since this stage reconstructs reduced tokens by

copying the outputs of their matched tokens, the resulting error is fundamentally determined by how closely the matched tokens’ outputs represent the original outputs of the reduced ones. Therefore, to effectively minimize this error, it is imperative that token matching reflects the true similarities of $\mathbf{Y}$, ensuring that the reduction and recovery stages are aligned with the final reconstruction. However, existing methods directly inherit matching strategies from ViT-style TR, which rely on input token similarity. This leads to a fundamental misalignment, making these approaches sub-optimal for minimizing recovery error.

4 Method

4.1 Output-Similarity-Aware Token Reduction

Motivation. To minimize the recovery error $\mathcal{L}_{\text{out}}$, the token matching must be performed using the output token similarity. However, since the dense output Y is only available after completing the full computation, we cannot directly perform token matching using Y . Therefore, a practical approach requires finding a proxy that can predict the output token similarity as closely as possible. To address this, we exploit the temporal consistency inherent in DMs: since activations remain similar across adjacent timesteps, we assume that the output token similarity is likewise preserved between neighboring timesteps.

To verify this hypothesis, we first define **Top- k Pair Match Rate (PMR)** as a metric that quantifies the alignment with the golden token matching results derived from the current-step output Y . Specifically, PMR measures the proportion of source tokens whose predicted Top- k destination set contains the single golden destination token. Given the competitive similarity scores among top candidates, this Top- k criterion provides a reasonable margin for error compared to a strict 1:1 match. The PMR definition is given as follows:

$$\text{PMR}_{\text{Top-}k}(t, b, \Delta) = \frac{1}{|S|} \sum_{s \in S} \mathbb{I} \left[\left| T_{t,b}^{(1)}(s) \cap T_{t-\Delta,b}^{(k)}(s) \right| > 0 \right]. \quad (4)$$

Here, S is the set of source tokens s , while $t \in \{0, \dots, T-1\}$ and $b \in \{0, \dots, B-1\}$ denote the timestep and block index, respectively. $T_{t,b}^{(k)}(s)$ represents the set of Top- k destination candidates for token s at timestep t and block b , and $\mathbb{I}[\cdot]$ is the indicator function. The parameter Δ determines the comparison target for evaluating its alignment with the golden matching results:

- $\Delta = 0$: Matching results derived from the current-step input.
- $\Delta \geq 1$: Matching results derived from the prior output at step $t - \Delta$.

Leveraging the PMR metric, Fig. 4a presents the timestep-wise average of $\text{PMR}_{\text{Top-}5}$ calculated over 50 prompts. The figure compares the matching consistency of the current input ($\Delta = 0$) and prior outputs ($\Delta \geq 1$) against the golden matching results, with the shaded regions visualizing the distribution across prompts. As shown in the figure, the PMR values of prior outputs are

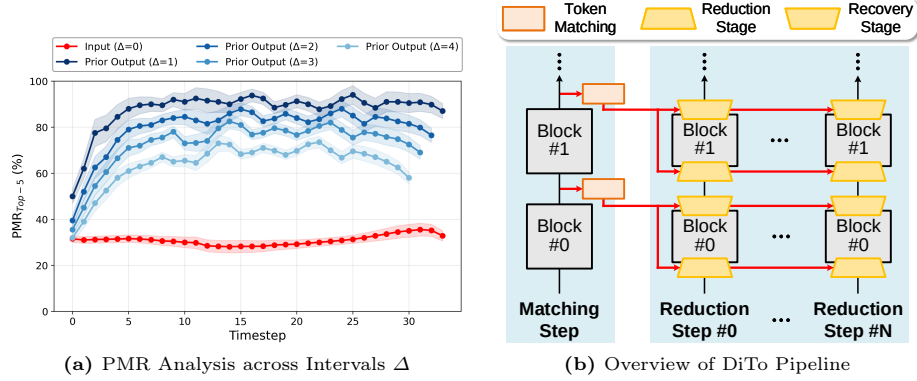


Fig. 4: Quantitative analysis of token alignment and DiTo pipeline. (a) PMR of prior outputs ($\Delta \geq 1$) remains consistently higher than that of the input ($\Delta = 0$), despite a gradual decline as Δ increases. (b) Leveraging this observation, DiTo performs token matching using the prior output in a single Matching step and reuses the matching results across subsequent Reduction steps to maximize efficiency.

significantly higher than those of the current input across most timesteps, validating our assumption that golden matching results are highly correlated with the results from prior steps rather than the current input. Based on these observations, we propose *Output-Similarity-Aware Token Reduction*, which performs token matching using the prior output to exploit the high temporal consistency of matching results across steps.

Proposed DiTo Pipeline. DiTo introduces a novel pipeline that decouples token matching from the reduction/recovery stage, which are originally handled within a single pipeline in conventional TR methods. Under this architecture, the timesteps of the diffusion process are categorized into two distinct types based on their roles. First, in the **Matching step**, the model performs full computation and utilizes the outputs for token matching to establish the token correspondences. Second, in the **Reduction step**, the model executes reduced computation by performing the reduction and recovery stages using the correspondences derived from the preceding Matching step. While DiTo maintains matching results throughout the Reduction steps, the associated memory overhead is negligible, as the stored information consists of compact index metadata rather than high-dimensional token features.

Built upon this design, the overall pipeline of DiTo is illustrated in Fig. 4b. The process consists of a single Matching step followed by a series of multiple Reduction steps. The rationale for this interleaved structure is grounded in the analysis presented in Fig. 4a; our results demonstrate that the PMR remains robust even as the reuse interval increases, indicating that matching results maintain a valid correlation over several subsequent timesteps. Consequently, rather than requiring a dedicated Matching step for every Reduction step, we leverage these high PMR values to reuse a single matching result across multiple

Algorithm 1: Assign Matching/Reduction Steps from Max Intervals ($\Delta_t^{\max}$)

Input: Max intervals $\{\Delta_t^{\max}\}_{t=0}^{T-1}$
Output: Lists $\mathcal{T}_{\text{match}}$ and $\mathcal{T}_{\text{reduce}}$

```

 $\mathcal{T}_{\text{match}} \leftarrow []$  ; // list of Matching steps
 $\mathcal{T}_{\text{reduce}} \leftarrow []$  ; // list of Reduction steps
 $m \leftarrow 0$  ; // Index of latest Matching step
for  $t \leftarrow 0$  to  $T - 1$  do
    if  $t = 0$  then
        // Initialize for the timestep 0
        Append 0 to  $\mathcal{T}_{\text{match}}$ ;
         $m \leftarrow 0$ 
    else
         $\Delta_{t+1} = (t + 1) - m$  ; // interval at  $t + 1$ 
        if  $\Delta_{t+1} > \Delta_{t+1}^{\max}$  then
            Append  $t$  to  $\mathcal{T}_{\text{match}}$ ;
             $m \leftarrow t$ ;
        else
            Append  $t$  to  $\mathcal{T}_{\text{reduce}}$ ;
// Keep only the last timestep in consecutive Matching steps
 $\mathcal{T}_{\text{match}} \leftarrow \text{CollapseConsecutive}(\mathcal{T}_{\text{match}})$ ;
return  $\mathcal{T}_{\text{match}}, \mathcal{T}_{\text{reduce}}$ ;

```

subsequent Reduction steps. This strategy maximizes computational efficiency while preserving high generation quality.

4.2 PMR-guided Interval Scheduling

While DiTo enhances efficiency through matching reuse, increasing the interval Δ leads to a decline in PMR, which can degrade generation quality. Furthermore, since PMR values fluctuate across different timesteps, determining the optimal reuse interval for each timestep is critical.

To address this, we propose **PMR-guided Interval Scheduling**, which utilizes the PMR metric as a guideline for determining the optimal matching schedule in an offline manner. The primary objective of this strategy is to maintain the PMR values above a predefined threshold τ throughout the generation process while maximizing efficiency. To achieve this, the transition between Matching and Reduction steps is dynamically determined by identifying the maximum interval $\Delta_t^{\max}$ at each timestep that satisfies the PMR criterion. Specifically, we first define the timestep-wise average PMR (PMR) as

$$\overline{\text{PMR}}_{\text{Top-}k}(t, \Delta) = \frac{1}{B} \sum_{b=0}^{B-1} \text{PMR}_{\text{Top-}k}(t, b, \Delta), \quad (5)$$

and then construct the valid interval set at each timestep t

$$\mathcal{D}_t = \left\{ \Delta \in \{1, \dots, T-1\} \mid \overline{\text{PMR}}_{\text{Top-}k}(t, \Delta) \geq \tau \right\}. \quad (6)$$

The maximum allowable interval ($\Delta_t^{\max}$) at each timestep t is chosen as

$$\Delta_t^{\max} = \begin{cases} \max \mathcal{D}_t, & \text{if } \mathcal{D}_t \neq \emptyset, \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

Given the max intervals $\Delta_t^{\max}$, Algorithm 1 assigns each timestep to either a Matching step or a Reduction step. The core scheduling rule follows a one-step look-ahead strategy: to determine whether timestep t should be a Matching or Reduction step, we evaluate the condition for the subsequent timestep $t+1$. Specifically, we calculate the expected interval at the next step as $\Delta_{t+1} = (t+1) - m$, where m denotes the index of the latest Matching Step. This value is then compared against the allowable limit $\Delta_{t+1}^{\max}$ to determine whether a new Matching step is required. If Δ_{t+1} exceeds the $\Delta_{t+1}^{\max}$, timestep t is designated as a Matching step to refresh the m ; otherwise, it is assigned as a Reduction step. Finally, to minimize redundant token matching operations, if Matching steps occur consecutively, we retain only the final timestep within the sequence; this is because matching results from preceding steps are overwritten by the subsequent update, making earlier computations unnecessary.

Since PMR primarily reflects the temporal consistency of matching reliability across denoising timesteps, we observe that it remains stable across different prompt sets, CFG scales, and random seeds. Therefore, for a fixed number of denoising timesteps T , the calibrated schedule can be successfully reused across diverse conditions without requiring per-condition re-scheduling. We provide the corresponding sensitivity analysis in the Appendix.

4.3 Frequency-aware Token Matching

Motivation. To improve efficiency, DiTo reuses the matching results in a single Matching step across multiple Reduction steps, reducing the frequency of token matching operations. However, as shown in the top-left heatmap of Fig. 5, this repeated reuse amplifies the imbalance in token selection frequency for reduction. Specifically, when a matching result is reused over d timesteps, the disparity in selection counts between frequently and rarely selected tokens expands in proportion to d . This causes reduction targets to persistently concentrate on a small subset of tokens, leading to the accumulation of localized approximation errors. These errors manifest as visible blocking artifacts, which, as illustrated in Fig. 5a, emerge at the exact spatial locations corresponding to the high-intensity regions in the heatmap.

To mitigate this issue, we propose *Frequency-aware Token Matching*, which augments the original token matching stage with a frequency-based penalty to alleviate the imbalance in the selection frequency. By penalizing tokens that

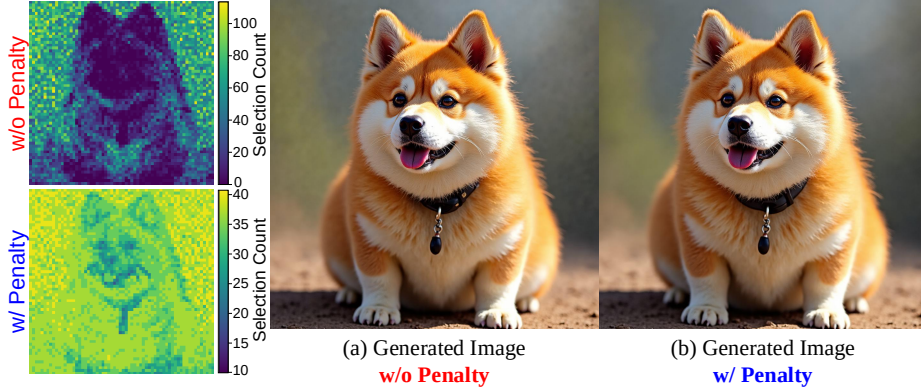


Fig. 5: Impact of token selection frequency in TR on image quality. (a) In naïve DiTo, repeated reuse of matching results makes reduction targets concentrate on a small set of spatial tokens (top-left heatmap), leading to visible blocking artifacts. (b) With our frequency-based penalty, the imbalance in selection frequency is alleviated (bottom-left heatmap). This reduces the accumulation of localized recovery errors and successfully mitigates blocking artifacts while maintaining the token reuse efficiency.

have been frequently selected as reduction targets, this method prevents excessive error accumulation at specific spatial locations. Consequently, it effectively mitigates blocking artifacts while maintaining the computational efficiency of token reuse.

Formulation. To ensure a scale-invariant penalty for every matching, we normalize the penalty magnitude using the similarity interval $\Delta s = \max(\hat{s}) - \min(\hat{s})$ from Eq. (2). This accounts for the fact that the absolute scale of similarity scores varies across different matching instances, providing a consistent reference range based on the distribution of candidate scores $\hat{s}$. Furthermore, considering that only a fraction r of tokens are selected during the matching stage, we define the baseline penalty scale as $r \cdot \Delta s$. The final penalty strength is controlled by a hyperparameter λ . We maintain a token selection history vector $\mathbf{C} \in \mathbb{R}^N$ over all N tokens, where C_i stores the cumulative number of times token i has been selected as a reduction target. Since the source token indices vary across token matching operations, we gather the history values for the current source set $\mathcal{S}$ as $\mathbf{C}[\mathcal{S}] \in \mathbb{R}^S$. We then augment the original token matching process by subtracting a frequency-based penalty $\mathbf{p}$ from the candidate similarity scores $\hat{s}$:

$$\mathbf{p} = \lambda \cdot r \cdot \Delta s \cdot \mathbf{C}[\mathcal{S}], \quad \hat{s}^{\text{pen}} = \hat{s} - \mathbf{p}. \quad (8)$$

As a result, as shown in the bottom-left heatmap of Fig. 5, the proposed penalty alleviates the imbalance in selection frequency. By reducing the accumulation of localized errors, it effectively mitigates blocking artifacts, as illustrated in Fig. 5b. Additional visualizations demonstrating this effect are provided in the Appendix.

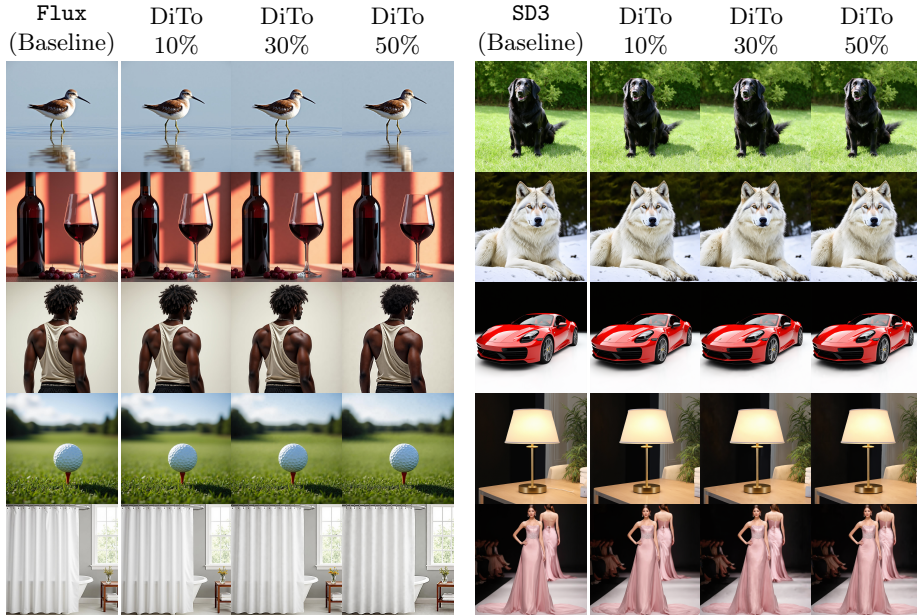


Fig. 6: Visualization results of DiTo under varying reduction ratios. As the token reduction ratio increases, DiTo consistently maintains high visual fidelity and structural consistency.

5 Experiments

5.1 Experimental Settings

Models and Benchmarks. We evaluate DiTo on the text-to-image task using two representative DiT models: FLUX.1-dev (Flux) [15] with 35 sampling steps and stable-diffusion-3-medium (SD3) [6] with 50 sampling steps. All experiments are conducted with the Diffusers framework [31], generating images at 1024×1024 resolution for evaluation, and FlashAttention2 [5] is applied to all models for efficient attention computation. We use the 1,000 class names from the ImageNet-1k dataset [35] as prompts, and generate a total of 3,000 images using three random seeds for the main experiments. For the ablation study, we randomly sample 100 images per seed, resulting in 300 images in total. Latency is measured on an RTX 6000 Ada GPU [28], averaged over 100 generated images.

Evaluation Metrics. We use FID [11], CLIP [10], PSNR, SSIM [39], and LPIPS [45] as evaluation metrics. FID (Fréchet Inception Distance) measures the distance between the distributions of generated and real images in the Inception feature space, capturing overall distributional quality and diversity. We compute FID using the ImageNet-1k ground-truth images as the reference set. CLIP score measures text-image alignment based on the similarity between generated image and text embeddings. In addition, we evaluate perceptual recon-

Table 1: Performance comparison of DiTo against prior methods for generating 1024×1024 images on FLUX (35 sampling steps). Best values are highlighted.

Ratio	Method	FID↓	PSNR↑	SSIM↑	LPIPS↓	CLIP↑	Latency↓	Δ ↓
Baseline	Flux	31.62	–	–	–	23.71	26.64	0%
0.25	DiTo	31.95	25.21	0.8552	0.2385	23.85	24.25	-9.0%
	ToMeSD	33.76	21.59	0.7817	0.3395	23.93	24.53	-7.9%
	ToFu	33.75	21.70	0.7851	0.3354	23.94	24.47	-8.1%
	SiTo	32.30	22.39	0.7965	0.3462	23.93	25.25	-5.2%
	ToMA	32.03	22.97	0.8112	0.2716	23.78	24.28	-8.9%
0.50	DiTo	33.47	22.04	0.7757	0.3329	24.01	21.69	-18.6%
	ToMeSD	90.00	19.73	0.6172	0.5384	23.04	22.00	-17.4%
	ToFu	83.79	19.81	0.6328	0.5302	23.05	22.02	-17.3%
	SiTo	61.05	20.17	0.6503	0.5245	23.45	22.54	-15.4%
	ToMA	43.71	20.49	0.6936	0.4386	23.90	21.62	-18.8%

struction fidelity using PSNR, SSIM, and LPIPS. PSNR (Peak Signal-to-Noise Ratio) quantifies pixel-level reconstruction error, SSIM (Structural Similarity) evaluates structural similarity, and LPIPS measures perceptual differences in a pretrained deep feature space. For these three metrics, we use the generated images from the vanilla model as the reference.

Baselines. We evaluate DiTo against an extensive set of representative TR methods, including ToMeSD [2], ToFu [13], SiTo [44], and ToMA [23]. Although ToMeSD, ToFu, and SiTo were originally proposed for U-Net models, they fundamentally operate by manipulating token similarity within a general TR pipeline consisting of matching, reduction, and recovery. From this perspective, their core mechanisms are not tied to a specific backbone architecture and can be naturally applied to DiTs. Accordingly, we include them as baselines by adapting their core components to the DiTs. Detailed implementations are provided in the Appendix.

5.2 Experimental Results

Quantitative Results. Tab. 1 presents quantitative comparisons between DiTo and existing TR methods (ToMeSD, ToFu, SiTo, and ToMA) on **Flux**. Overall, DiTo consistently outperforms all methods in terms of generation quality across multiple metrics, including FID, PSNR, SSIM, and LPIPS. Notably, DiTo demonstrates substantial improvements in perceptual reconstruction fidelity, achieving PSNR gains of 2.24–3.62 dB at a 0.25 ratio and 1.55–2.31 dB at a 0.50 ratio. These gains are also reflected in other key metrics, including SSIM and LPIPS. These results align with the design objective of DiTo, which explicitly aims to minimize recovery error by incorporating output token similarity

Table 2: Performance comparison of DiTo against prior methods for generating 1024×1024 images on SD3 (50 sampling steps). Best values are highlighted.

Ratio	Method	FID↓	PSNR↑	SSIM↑	LPIPS↓	CLIP↑	Latency↓	Δ ↓
Baseline	SD3	27.67	—	—	—	24.67	9.32	0%
0.25	DiTo	27.83	26.83	0.9160	0.1492	24.77	8.63	-7.4%
	ToMeSD	28.42	23.32	0.8697	0.2045	24.74	8.82	-5.3%
	ToFu	27.99	23.83	0.8835	0.1922	24.72	8.87	-4.9%
	SiTo	27.92	23.01	0.8682	0.2211	24.79	9.30	-0.2%
	ToMA	28.47	22.93	0.8624	0.2029	24.63	9.08	-2.5%
0.50	DiTo	28.89	23.16	0.8510	0.2577	24.87	7.79	-16.4%
	ToMeSD	31.45	21.22	0.8090	0.3139	24.80	8.00	-14.2%
	ToFu	31.89	21.21	0.8112	0.3187	24.79	8.05	-13.6%
	SiTo	33.13	21.23	0.8046	0.3170	24.82	8.42	-9.7%
	ToMA	33.37	20.91	0.8049	0.3053	24.62	8.17	-12.3%

into the TR process. As these metrics directly measure perceptual fidelity with respect to the vanilla model output, the consistent gains confirm that DiTo effectively achieves its intended objective. Furthermore, DiTo exhibits exceptional robustness to the reduction ratio; while existing methods suffer from substantial degradation in FID scores at a 0.50 ratio (ranging from 43.71 to 90.00), DiTo maintains a significantly more stable FID of 33.47.

Regarding semantic alignment, DiTo achieves CLIP scores (23.85–24.01) comparable to or even slightly exceeding the vanilla model (23.71), indicating that text-to-image consistency is well preserved. In terms of efficiency, by reusing matching results, DiTo achieves up to an 18.6% latency reduction (21.69s), which is comparable to the runtime of ToMA—the fastest among the compared TR methods—while providing superior generation quality. Tab. 2 reports the results on SD3, where similar trends are consistently observed. These results demonstrate that DiTo generalizes well across different models and remains robust even with fewer diffusion timesteps (e.g., 35 steps in Flux).

Qualitative Comparison. Fig. 1 presents qualitative comparisons between DiTo and existing token reduction methods (ToMeSD, ToFu, SiTo, and ToMA) on Flux and SD3. Under the equivalent FLOPs reduction, DiTo consistently preserves higher visual fidelity and structural consistency across both models, whereas prior methods exhibit noticeable degradation. Additional qualitative comparisons are provided in the Appendix. Fig. 6 further presents visualization results of DiTo across different reduction ratios on both models, demonstrating that image quality remains stable even as the reduction ratio increases. Additional visualizations are provided in the Appendix.

Ablation & Others. To validate our design decisions, we conduct comprehensive ablation studies in the Appendix, evaluating the impact of the PMR threshold τ and penalty strength λ to identify the optimal balance between efficiency and quality. Furthermore, we examine the choice of reduction strategies (e.g., pruning/merging) and various similarity metrics. We also provide a comparative analysis of FLOPs and memory consumption across different reduction methods. Our profiling confirms that while DiTo maintains matching results throughout the Reduction steps, the associated memory overhead is negligible, as the stored information consists of compact index metadata rather than high-dimensional token features.

Additionally, we provide further evaluations to demonstrate the robustness and generalizability of DiTo, along with a comparison against a representative caching-based acceleration method in the Appendix. First, we evaluate DiTo using additional LLM-based metrics from two complementary perspectives: text alignment and image quality. The results show that DiTo consistently outperforms existing TR baselines in both metrics, further confirming that DiTo preserves semantic alignment and perceptual quality. Second, we verify the architectural generalizability of DiTo by applying it beyond MM-DiT architectures to both U-Net (SD1.5 [34]) and standard DiT (DiT [30]) architectures. Lastly, we compare DiTo with ToCa [50], a representative caching-based acceleration method. While ToCa slightly improves FID, DiTo achieves comparable latency with substantially lower memory overhead, highlighting the scalability advantage of metadata-based token reduction over feature caching.

6 Conclusion

In this work, we introduce **DiTo**, a novel paradigm that shifts toward output-centric token reduction to explicitly minimize recovery error. To achieve this, DiTo leverages the temporal consistency of DMs and establishes token correspondences using prior-step similarities as a proxy at a Matching step, and reuses the resulting correspondences across multiple subsequent Reduction steps. To optimize this scheduling, we propose PMR-guided Interval Scheduling to determine the optimal matching frequency. Additionally, to mitigate localized approximation errors and blocking artifacts from repeated reuse, we introduce Frequency-aware Token Matching with a selection frequency penalty. Extensive experiments demonstrate that DiTo effectively breaks the quality-efficiency trade-off, achieving a superior Pareto frontier compared to existing methods.

7 Acknowledgement

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (IITP-2026-RS-2020-II201847, Information Technology Research Center; IITP-2026-RS-2023-00256472, Graduate School of AI Semiconductor).

References

1. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your ViT but faster. In: *Int. Conf. Learn. Represent.* (2023)
2. Bolya, D., Hoffman, J.: Token merging for fast stable diffusion. In: *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.* (2023)
3. Chen, J., Ge, C., Xie, E., Wu, Y., Yao, L., Ren, X., Wang, Z., Luo, P., Lu, H., Li, Z.: Pixart- σ : Weak-to-strong training of diffusion transformer for 4k text-to-image generation (2024)
4. Chen, Y., Ma, Z., Yang, C., An, Z., Zhang, Y.: Accelerating diffusion models via parallel denoising. In: *ACM Int. Conf. Multimedia* (2025)
5. Dao, T.: FlashAttention-2: Faster attention with better parallelism and work partitioning. In: *Int. Conf. Learn. Represent.* (2024)
6. Esser, P., Kulal, S., Andreas, B., Enright, A., Sheynin, J., Sauer, A., Chen, D., Podell, D., Evans, D., Brack, M., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: *Int. Conf. Mach. Learn.* (2024)
7. Fayyaz, M., Koohpayegani, S.A., Jafari, F.R., Sengupta, S., Joze, H.R.V., Sommerlade, E., Pirsiavash, H., Gall, J.: Adaptive token sampling for efficient vision transformers. In: *Eur. Conf. Comput. Vis.* (2022)
8. Guo, B., Tang, S., Zeng, C., Shen, Z.: Mosaicdiff: Training-free structural pruning for diffusion model acceleration reflecting pretraining dynamics. In: *Int. Conf. Comput. Vis.* (2025)
9. He, Y., Liu, L., Liu, J., Wu, W., Zhou, H., Zhuang, B.: Ptqd: Accurate post-training quantization for diffusion models. In: *Adv. Neural Inform. Process. Syst.* (2023)
10. Hessel, J., Holtzman, A., Forbes, M., Bras, R.L., Choi, Y.: Clipscore: A reference-free evaluation metric for image captioning. In: *Conf. Empirical Methods Natural Lang. Process.* (2021)
11. 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. In: *Adv. Neural Inform. Process. Syst.* (2017)
12. Kim, B.K., Song, H.K., Castells, T., Choi, S.: Bk-sdm: A lightweight, fast, and cheap version of stable diffusion. In: *Eur. Conf. Comput. Vis.* (2024)
13. Kim, M., Gao, S., Hsu, Y.C., Shen, Y., Jin, H.: Token fusion: Bridging the gap between token pruning and token merging. In: *IEEE Winter Conf. Appl. Comput. Vis.* pp. 1372–1381 (2024)
14. Kong, Z., Dong, P., Ma, X., Meng, X., Niu, W., Sun, M., Shen, X., Yuan, G., Ren, B., Tang, H., et al.: Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In: *Eur. Conf. Comput. Vis.* (2022)
15. Labs, B.F.: Flux. <https://github.com/black-forest-labs/flux> (2024), accessed: 2026-03-05
16. Lee, Y., Park, K., Cho, Y., Lee, Y., Hwang, S.: Koala: Empirical lessons toward memory-efficient and fast diffusion models for text-to-image synthesis. In: *Adv. Neural Inform. Process. Syst.* (2024)
17. Lefaudeux, B., Massa, F., Liskovich, D., Xiong, W., Caggiano, V., Naren, S., Xu, M., Hu, J., Tintore, M., Zhang, S., Labatut, P., Haziza, D., Wehrstedt, L., Reizenstein, J., Sizov, G.: xformers: A modular and hackable transformer modelling library. <https://github.com/facebookresearch/xformers> (2022), accessed: 2026-03-05
18. Li, X., Liu, Y., Lian, L., Yang, H., Dong, Z., Kang, D., Zhang, S., Keutzer, K.: Q-diffusion: Quantizing diffusion models. In: *Int. Conf. Comput. Vis.* (2023)

19. Liang, Y., Ge, C., Tong, Z., Song, Y., Wang, J., Xie, P.: Evit: Expediting vision transformers via token reorganizations. In: *Int. Conf. Learn. Represent.* (2022)
20. Liu, X., Li, Z., Xiao, J., Chen, M., Li, J., Gu, Q.: Eda-dm: Enhanced distribution alignment for post-training quantization of diffusion models (2024), <https://arxiv.org/abs/2401.04585>, accessed: 2026-03-05
21. Lu, Cheng, e.a.: Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In: *Adv. Neural Inform. Process. Syst.* (2022)
22. Lu, Cheng, e.a.: Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research* (2025)
23. Lu, W., Zheng, S., Xia, Y., Wang, S.: ToMA: Token merge with attention for diffusion models. In: *Int. Conf. Mach. Learn.* (2025)
24. Luo, S., Tan, Y., Huang, L., Li, J., Zhao, H.: Latent consistency models: Synthesizing high-resolution images with few-step inference (2023), <https://arxiv.org/abs/2310.04378>
25. Marin, D., Chang, J.R., Ranjan, A., Prabhu, A., Rastegari, M., Tuzel, O.: Token pooling in vision transformers for image classification. In: *IEEE Winter Conf. Appl. Comput. Vis.* (2023)
26. Meng, C., Rombach, R., Gao, R., Kingma, D.P., Ermon, S., Ho, J., Salimans, T.: On distillation of guided diffusion models. In: *IEEE Conf. Comput. Vis. Pattern Recog.* (2023)
27. Meng, X., Li, X., Wang, Y., Wu, X., Zhang, Y., Sun, J.: Adavit: Adaptive vision transformers for efficient image recognition. In: *IEEE Conf. Comput. Vis. Pattern Recog.* (2022)
28. NVIDIA Corporation: Nvidia rtx 6000 ada generation. <https://www.nvidia.com/en-us/products/workstations/rtx-6000/> (2023), accessed: 2026-03-05
29. Pan, X., Ge, C., Lu, R., Song, S., Huang, G., Wang, Z., Huang, Z.: Ia-red²: Interpretability-aware redundancy reduction for vision transformers. In: *Adv. Neural Inform. Process. Syst.* (2021)
30. Peebles, W., Xie, S.: Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748* (2022)
31. von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., Nair, D., Paul, S., Berman, W., Xu, Y., Liu, S., Wolf, T.: Diffusers: State-of-the-art diffusion models. <https://github.com/huggingface/diffusers> (2022), accessed: 2026-03-05
32. Proust, M., Martyna Poreba, Michal Szczepanski, K.H.: Step: Supertoken and early-pruning for efficient semantic segmentation. In: *Int. Conf. Comput. Vis. Theory Appl.* (2025)
33. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.J.: Dynamicvit: Efficient vision transformers with dynamic token sparsification. In: *Adv. Neural Inform. Process. Syst.* (2021)
34. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: *IEEE Conf. Comput. Vis. Pattern Recog.* (2022)
35. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* (2015)
36. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. In: *Int. Conf. Learn. Represent.* (2022)
37. Smith, E., Saxena, N., Saha, A.: Todo: Token downsampling for efficient generation of high-resolution images. In: *IJCAI* (2024)

38. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: Int. Conf. Learn. Represent. (2021)
39. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* (2004)
40. Yang, Y., Yue Zhou, Xiaofang Hu, S.D.: K-feature fusion token merging for vision transformer. *Expert Systems with Applications* (2025)
41. Yin, H., Vahdat, A., Alvarez, J.M., Mallya, A., Kautz, J., Molchanov, P.: A-ViT: Adaptive Tokens for Efficient Vision Transformer. In: *IEEE Conf. Comput. Vis. Pattern Recog.* (2022)
42. Yin, T., Gharbi, M., Park, T., Zhang, R., Shechtman, E., Durand, F., Freeman, W.T.: Improved distribution matching distillation for fast image synthesis. In: *Adv. Neural Inform. Process. Syst.* (2024)
43. Zhan, Z., Wu, Y., Gong, Y., Meng, Z., Kong, Z., Yang, C., Wang, Y.: Fast and memory-efficient video diffusion using streamlined inference. In: *Adv. Neural Inform. Process. Syst.* (2024)
44. Zhang, E., Tang, J., Ning, X., Zhang, L.: Training-free and hardware-friendly acceleration for diffusion models via similarity-based token pruning. In: *AAAI* (2025)
45. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: *IEEE Conf. Comput. Vis. Pattern Recog.* (2018)
46. Zhao, W., Bai, L., Rao, Y., Zhou, J., Lu, J.: Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. In: *Adv. Neural Inform. Process. Syst.* (2023)
47. Zheng, X., Liu, X., Bian, Y., Ma, X., Zhang, Y., Wang, J., Qin, H.: Bidm: Pushing the limit of quantization for diffusion models. In: *Adv. Neural Inform. Process. Syst.* (2024)
48. Zhou, Z., Chen, D., Wang, C., Chen, C.: Fast ode-based sampling for diffusion models in around 5 steps. In: *IEEE Conf. Comput. Vis. Pattern Recog.* (2024)
49. Zhu, J., Wang, H., Su, M., Wang, Z., Wang, H.: Obs-diff: Accurate pruning for diffusion models in one-shot. In: *Int. Conf. Comput. Vis.* (2026)
50. Zou, C., Liu, X., Liu, T., Huang, S., Zhang, L.: Accelerating diffusion transformers with token-wise feature caching. In: *Int. Conf. Learn. Represent.* (2025)