COMPENSATING TOKEN MERGING FOR EFFICIENT DIFFUSION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Token merging approaches accelerate diffusion transformers (DiTs) by aggregating redundant tokens into representative ones to reduce the quadratic cost of attention. Recent approaches further employ advanced selection criteria to identify redundant tokens for aggregation. Despite their effectiveness, they still suffer from significant merging errors, the discrepancies between full-token and merged-token outputs, leading to blurred textures and local artifacts. However, directly computing these errors requires attention over the full tokens, which negates the efficiency gain of token merging. To address this, we propose ComE-ToMe, a novel framework that compensates for the merging errors efficiently. Specifically, we introduce calibrated window attention (CWA) that estimates the errors within local neighborhoods to restore the lost details. By computing attention within local windows while incorporating outside-window tokens through their averaged keys, CWA effectively captures fine details without the cost of full attention. We further introduce compensation-aware merging (CAM), which adapts token merging for CWA by preserving boundary tokens and using cumulative CFG importance to stably protect salient regions across denoising steps. Extensive experiments across various DiT architectures demonstrate that ComE-ToMe consistently outperforms prior approaches while maintaining efficiency. We will make our code publicly available online upon acceptance.

1 INTRODUCTION

Diffusion models (Ho et al., 2020; Song et al., 2021) have established new standards for generating high-fidelity images (Rombach et al., 2022; Saharia et al., 2022; Esser et al., 2024; Chen et al., 2024; Xie et al., 2025) and videos (Ho et al., 2022; Kong et al., 2024; Yang et al., 2025b; Wan et al., 2025). Recently, Diffusion Transformers (DiTs) (Peebles & Xie, 2023) further improve generation quality by adopting the attention mechanism (Vaswani et al., 2017; Dosovitskiy et al., 2021) that partitions visual data into a sequence of tokens, and feeds them into attention layers to capture long-range dependencies. However, attention mechanism scales quadratically with the number of tokens, incurring substantial cost at high resolutions. The cost is repeated at every denoising step, leading to substantial computational overhead and higher inference latency.

To mitigate this, recent research has explored to reduce the number of tokens in DiTs for efficient inference. The seminal work of ToMeSD (Bolya & Hoffman, 2023) adapts token merging (Bolya et al., 2023) for diffusion architectures, reducing the number of tokens by averaging tokens into representative ones. While token reduction methods in Vision Transformers (ViTs) (Pan et al., 2021; Rao et al., 2021; Xu et al., 2022; Liang et al., 2022; Yin et al., 2022; Kong et al., 2022; Long et al., 2023; Wei et al., 2023; Liu et al., 2024; Lee et al., 2024; Tran et al., 2024; Lee et al., 2025a) prune or merge tokens permanently to extract a global representation for recognition, DiTs require the original number of tokens to provide spatial noise estimates for dense prediction. To bridge this gap, ToMeSD introduces an unmerging operation. It restores the number of tokens by duplicating merged tokens back to their original positions. That is, the tokens are merged before being fed into modules (*i.e.*, attention layers), and then unmerged afterward. However, this merge-compute-unmerge pipeline attenuates fine-grained details, with blurry visual outputs or local artifacts (Fig. 1). The degradation stems from the discrepancies between outputs computed with the full token sequence and those from merged tokens, which we define as *merging errors*. These errors fundamentally limit preserving

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



Figure 1: Visual comparison using PixArt-α (Chen et al., 2024). Previous methods with a reduction ratio of 0.5, including ToMeSD (Bolya & Hoffman, 2023), ToMA (Lu et al., 2025b), and Imp-ToMe (Wu et al., 2025), show local artifacts and loss of fine-grained details. In contrast, ComE-ToMe faithfully restores fine textures and local structures compared to the baseline by compensating for merging errors.

feature representations in existing methods (Fang et al., 2025; Wang et al., 2024; Lu et al., 2025b; Wu et al., 2025), regardless of their merging strategies.

In this paper, we propose ComE-ToMe, a novel framework for DiTs that **Com**pensates for **E**rrors in **To**ken **Me**rging to restore fine-grained details of duplicated token features. The key challenge of compensating for merging errors lies in estimating the attention output with high accuracy and minimal computational overhead, as the attention mechanism plays a central role in forming fine-grained visual details in DiTs. However, calculating the exact error requires attention over the full token sequence, which negates the benefit of token merging. To address this, we propose to approximate the merging errors effectively and efficiently using a calibrated window attention (CWA) technique. CWA estimates the merging error by computing local window attention, while adjusting the attention score to incorporate the influence of tokens outside the window. This approach is motivated by inherent locality in DiTs (Rissanen et al., 2023; Kwon et al., 2023; Qian et al., 2024; Lu et al., 2024; Kynkäänniemi et al., 2024; Kim et al., 2026), where fine-grained details are primarily driven by interactions among neighboring tokens (Dong et al., 2025). We also introduce a compensation-aware merging (CAM) strategy to avoid merging in regions where CWA cannot accurately estimate the merging error. Specifically, CAM preserves more tokens in regions likely to have large merging error (e.g., edges of local windows) and those with high cumulative importance over the denoising process.

We validate our approach across a range of modern DiT architectures, including DiT (Peebles & Xie, 2023), MMDiT (Esser et al., 2024), and VideoDiT (Wan et al., 2025). Experimental results show that ComE-ToMe consistently improves visual fidelity over existing methods without retraining. Our main contributions are summarized as follows:

- We present a frequency-domain analysis of existing token merging methods for DiTs, showing that they lose high-frequency components significantly, especially at later denoising timesteps.
- We propose ComE-ToMe, a training-free framework that consists of CWA, which compensates for merging errors to restore the lost high-frequency details, and CAM, which adapts token merging to CWA.
- Extensive experiments and comprehensive ablation studies demonstrate consistent improvements of our method across various diffusion architectures, while preserving efficiency.

2 RELATED WORK

2.1 EFFICIENT DIFFUSION MODELS

Diffusion models require significant computational cost mainly due to their large-scale network architectures and the iterative denoising process. To address this, several works have explored improving the efficiency of diffusion models. For example, quantization methods (Li et al., 2023; Huang et al., 2024a; Yao et al., 2024; Lee et al., 2025b) reduce the bit-width of the weights and activations, and feature caching methods (Selvaraju et al., 2024; Liu et al., 2025a;b; Son et al., 2026) reuse intermediate features, which is based on the temporal redundancy in denoising steps. Distillation techniques (Salimans & Ho, 2022; Song et al., 2023) have also been widely studied to transfer the generation capability of a pretrained diffusion model to a model with fewer denoising steps. Despite their effectiveness, these approaches often require additional procedures or resources. Quantization methods often rely on calibration to determine quantization parameters, and feature caching methods require additional memory to store intermediate features. Distillation techniques usually involve a re-training stage, incurring substantial training costs. These limitations highlight the need for a method that reduces computational redundancy without introducing additional overhead, while preserving the behavior of pretrained diffusion models.

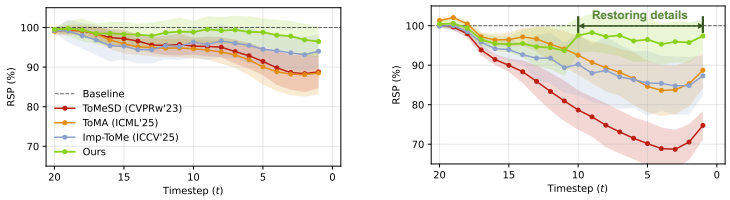
2.2 TOKEN REDUCTION

Token reduction has emerged as a promising approach for accelerating transformer architectures without finetuning or additional memory overhead. Early approaches for recognition tasks prune unimportant tokens (Pan et al., 2021; Rao et al., 2021; Xu et al., 2022; Liang et al., 2022; Yin et al., 2022; Kong et al., 2022) or merge redundant ones (Long et al., 2023; Wei et al., 2023; Liu et al., 2024; Lee et al., 2024; Tran et al., 2024; Lee et al., 2025a) based on attention scores or feature similarity. Since recognition models eventually aggregate token features into a single global representation (*e.g.*, via a [CLS] token or average pooling), these methods can permanently discard tokens across transformer layers. In contrast, dense prediction tasks such as image generation require the original number of tokens to predict noise estimates, limiting the direct application of token reduction methods to diffusion models. To resolve this, ToMeSD (Bolya & Hoffman, 2023) extends token merging to diffusion models by introducing an unmerging operation, where tokens are duplicated back to their original spatial positions after the attention computation. Within this merge-and-unmerge pipeline, subsequent approaches have developed more effective token selection criteria to identify redundant tokens for diffusion models. For instance, AT-EDM (Wang et al., 2024) uses self-attention maps to rank token importance and prunes tokens with low attention scores. Imp-ToMe (Wu et al., 2025) uses classifier-free guidance (CFG) (Ho & Salimans, 2021) to define token importance and merges tokens in regions with low CFG scores. SDTM (Fang et al., 2025) merges tokens on local similarity in early steps and global attention in later steps, exploiting the coarse-to-fine denoising property of diffusion models. Although token reduction methods for diffusion models propose advanced token selection strategies, they rely on the merge-compute-unmerge pipeline of ToMeSD. This pipeline inherently suffers from merging errors defined as discrepancies between the outputs computed with all tokens and those from reduced tokens, which degrades the quality of generation. While ToMA (Lu et al., 2025b) reformulates merging and unmerging operations as linear transformations that leverage GPU-efficient matrix multiplications, its primary objective is to minimize latency rather than reduce merging errors. As a result, ToMA exhibits image quality comparable to or worse than ToMeSD, particularly at high reduction ratios. To address this limitation, we introduce an error compensation approach that restores the fine-grained details lost by merging errors.

3 METHOD

3.1 PROBLEM SETUP

Given an input matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$ fed into the self-attention module, where we denote by N and d as the number of tokens and channels, respectively, token merging methods reduce the number of tokens by aggregating redundant tokens into compressed ones. Formally, they apply a merging matrix $\mathbf{M} \in \mathbb{R}^{N(1-r) \times N}$ to $\mathbf{X}$ to obtain a compressed representation $\bar{\mathbf{X}} = \mathbf{M}\mathbf{X} \in \mathbb{R}^{N(1-r) \times d}$, where we denote by $r \in (0, 1)$ the reduction ratio. The self-attention operator $f(\cdot)$ is then applied to the merged tokens, providing $f(\bar{\mathbf{X}}) \in \mathbb{R}^{N(1-r) \times d}$. An unmerging matrix $\mathbf{U} \in \mathbb{R}^{N \times N(1-r)}$ is then applied to map the merged outputs back to the original token positions, yielding $\mathbf{U}f(\bar{\mathbf{X}}) \in \mathbb{R}^{N \times d}$. We define



(a) Low-frequency band ($0 \leq \omega \leq \frac{\pi}{4}$). (b) High-frequency band ($\frac{3\pi}{4} \leq \omega \leq \pi$).

Figure 2: Frequency analysis using PixArt-α (Chen et al., 2024) on the COCO 2017 validation set (Lin et al., 2014) with $r = 0.5$. RSP of features across timesteps for (a) low-frequency and (b) high-frequency bands. Please see the text for details.

the resulting merging error relative to the original output as:

$$\Delta f = f(\mathbf{X}) - \mathbf{U} f(\bar{\mathbf{X}}). \tag{1}$$

The error term Δf serves as a compensation term for recovering the information loss induced by token merging. However, directly computing Δf requires evaluating $f(\mathbf{X})$ over the full set of tokens, which diminishes the efficiency gains of token merging. In the following subsection, we seek an efficient approximation of Δf without applying self-attention to the full set of tokens.

3.2 Motivation

The merging error Δf represents the discrepancy between $f(\mathbf{X})$ and $\mathbf{U} f(\bar{\mathbf{X}})$, but the type of information lost in Δf remains unclear. To identify it, we conduct a frequency-domain analysis. We visualize in Fig. 2 the relative spectrum power (RSP), which measures the spectrum of $\mathbf{U} f(\bar{\mathbf{X}})$ relative to that of $f(\mathbf{X})$, for prior merging methods (Bolya & Hoffman, 2023; Lu et al., 2025b; Wu et al., 2025) in the low-frequency ($0 \leq \omega \leq \frac{\pi}{4}$) and high-frequency ($\frac{3\pi}{4} \leq \omega \leq \pi$) ranges. We can see that low-frequency components are preserved across merging methods without significant degradation (Fig. 2(a)). In contrast, high-frequency components show a substantial drop, especially at later denoising timesteps (Fig. 2(b)).

This high-frequency loss is mainly due to inherent low-pass filtering behavior of token merging operation. The merging matrix $\mathbf{M}$ aggregates spatially adjacent tokens, and acts as a low-pass filter on $\mathbf{X}$, yielding $\bar{\mathbf{X}}$. Notably, the unmerging matrix $\mathbf{U}$ simply duplicates the attention outputs $f(\bar{\mathbf{X}})$, and thus cannot restore the token-level variations removed during merging. Motivated by this observation, we focus on recovering high-frequency components. To this end, we propose ComE-ToMe that restores fine-grained details efficiently (Fig. 3). Leveraging the well-known locality of diffusion models (Rissanen et al., 2023; Kwon et al., 2023; Qian et al., 2024; Lu et al., 2024; Kynkäänniemi et al., 2024; Kim et al., 2026) in later timesteps, we compensate for merging errors Δf within a local neighborhood around each token, which restores detail without re-introducing the cost of full attention.

3.3 ComE-ToMe

Calibrated Window Attention (CWA). Based on the motivation, we adopt window attention to compensate for merging errors efficiently. In particular, we reshape input tokens $\mathbf{X}$ into a 2D spatial grid and partition it into non-overlapping windows, each containing N_W tokens. Given a local window W, we define a local merging error Δf_W in the window as follows:

$$\Delta f_W = f_W(\mathbf{X}) - \mathbf{U} f_W(\bar{\mathbf{X}}). \tag{2}$$

Here, we refer to $f_W(\cdot)$ as the *local contribution* induced by the decomposition of $f(\mathbf{X})$, *i.e.*, $f_W(\mathbf{X}) + f_E(\mathbf{X})$, where f_W and f_E denote the attention contributions from tokens inside and outside the window, respectively. The local contribution $f_W(\mathbf{x})$ is defined as follows:

$$f_W(\mathbf{x}) = \frac{\sum_{j \in W} \exp(z_j) \mathbf{v}_j}{\sum_{i=1}^{N} \exp(z_i)}, \tag{3}$$

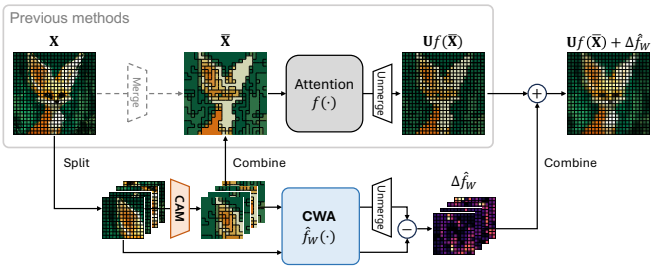


Figure 3: An overview of ComE-ToMe. Input tokens are first split into local windows. Both the original tokens and the tokens merged via CAM strategy are simultaneously fed into the CWA module. The difference between their outputs is then used to approximate the merging error.

where $z_i = \frac{1}{\sqrt{d}}\mathbf{q}\mathbf{k}_i^\top$ denotes the attention logit between the query vector $\mathbf{q}$ and the key vector $\mathbf{k}_i$ of the i-th token, and $\mathbf{v}_i$ denotes its value vector. Note that the local contribution $f_W(\mathbf{x})$ is different from the output of standard window attention $\tilde{f}_W(\mathbf{x})$, which is defined as:

$$\tilde{f}_W(\mathbf{x}) = \frac{\sum_{j\in W}\exp(z_j)\mathbf{v}_j}{\sum_{i\in W}\exp(z_i)}. \tag{4}$$

When comparing $f_W(\mathbf{x})$ and $\tilde{f}_W(\mathbf{x})$, their numerators use the same tokens in W, but different denominators. That is, the local contribution $f_W(\mathbf{x})$ is normalized over all tokens, whereas $\tilde{f}_W(\mathbf{x})$ is normalized only over tokens inside W. Without correcting this normalization gap, $\tilde{f}_W(\mathbf{x})$ would overemphasize the in-window attention scores, since excluding the outside-window terms reduces the denominator. To bridge this gap, we calibrate $\tilde{f}_W(\mathbf{x})$ by correcting its denominator toward the denominator of $f_W(\mathbf{x})$. Note that the calibration requires estimating the missing sum of attention scores contributed by tokens outside the window, which requires computing attention over all tokens. Instead, we approximate these attention scores using a single representative key for each external window, with an assumption that nearby tokens within the same external window tend to provide similar attention interactions with queries in a local window, as follows:

$$\bar{\mathbf{k}}_i = \frac{1}{N_W}\sum_{j\in E_i}\mathbf{k}_j. \tag{5}$$

where E_i denotes the i-th external window. Using Eq. (5), we approximate the outside window contribution to the attention denominator of Eq. (3) as

$$\sum_{i\in E}\exp(z_i) \approx N_W\sum_{i=1}^{L-1}\exp(\bar{z}_{E_i}), \quad \bar{z}_{E_i} = \frac{1}{\sqrt{d}}\mathbf{q}\bar{\mathbf{k}}_i^\top, \tag{6}$$

where $L = N/N_W$ denotes the number of non-overlapping windows, and $\{E_i\}_{i=1}^{L-1}$ are the windows outside W. Rather than summing the attention scores of all external tokens, we estimate the external score mass in Eq. (3) using the representative key of each external window. This modifies the denominator of window attention in Eq. (4) to account for the missing outside-window attention scores, yielding the calibrated local contribution:

$$\hat{f}_W(\mathbf{x}) = \frac{\sum_{j\in W}\exp(z_j)\mathbf{v}_j}{\sum_{j\in W}\exp(z_j) + N_W\sum_{i=1}^{L-1}\exp(\bar{z}_{E_i})}. \tag{7}$$

Using the approximated local contribution $\hat{f}_W(\cdot)$, we estimate the local merging error as:

$$\Delta\hat{f}_W = \hat{f}_W(\mathbf{X}) - \mathbf{U}\hat{f}_W(\bar{\mathbf{X}}). \tag{8}$$

Analysis of the key approximation. We analyze the external-window logit matrix to examine whether the keys in an external window can be represented by a single proxy as Eq. (5). Consider the logit matrix $\mathbf{Z}_i \in \mathbb{R}^{N_W\times N_W}$ between the queries in W and the keys in an external window E_i, whose (m,n)-th element is

$$z_{mn} = \frac{1}{\sqrt{d}}\mathbf{q}_m^\top\mathbf{k}_n. \tag{9}$$

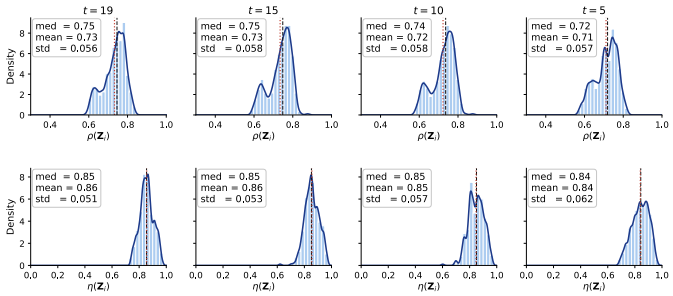


Figure 4: Analysis of the external-window logit matrix $\mathbf{Z}_i$. Top: rank-one dominance $\rho(\mathbf{Z}_i)$. Bottom: key-side uniformity $\eta(\mathbf{Z}_i)$. Across timesteps, $\mathbf{Z}_i$ is dominated by a leading singular component, and its leading right singular vector is close to the uniform direction.

Here, $\mathbf{q}_m$ denotes the m-th query in W, and $\mathbf{k}_n$ denotes the n-th key in E_i. First, we apply singular value decomposition to the logit matrix $\mathbf{Z}_i$ to examine its rank as follows:

$$\mathbf{Z}_i = \sum_j \sigma_j(\mathbf{Z}_i)\boldsymbol{\mu}_j\boldsymbol{\nu}_j^\top, \tag{10}$$

where $\sigma_j(\mathbf{Z}_i)$ is the j-th singular value, and $\boldsymbol{\mu}_j$ and $\boldsymbol{\nu}_j$ are the corresponding left and right singular vectors, respectively. We then measure the dominance of the largest singular value:

$$\rho(\mathbf{Z}_i) = \frac{\sigma_1^2(\mathbf{Z}_i)}{\sum_j \sigma_j^2(\mathbf{Z}_i)}. \tag{11}$$

As shown in the top row of Fig. 4, $\rho(\mathbf{Z}_i)$ exceeds 0.7 for most external windows, with per-timestep medians in $[0.72,\ 0.75]$. This indicates that the interaction between local queries and external-window keys is largely explained by a single rank-one component. Second, we examine the leading rank-one component $\sigma_1\boldsymbol{\mu}_1\boldsymbol{\nu}_1^\top$ more closely. This component is the outer product of two unit singular vectors, $\boldsymbol{\mu}_1$ on the query side and $\boldsymbol{\nu}_1$ on the key side. A notable pattern is that the key-side direction $\boldsymbol{\nu}_1$ is close to the uniform direction over the external window. This pattern is consistent with local redundancy within a small spatial window. Keys from the same external window tend to induce similar logit responses across the queries in W. In this case, the columns of $\mathbf{Z}_i$ become similar to one another, and the leading right singular vector $\boldsymbol{\nu}_1$ approaches the uniform direction over the key positions. To quantify this, we measure

$$\eta(\mathbf{Z}_i) = \left|\boldsymbol{\nu}_1^\top \frac{\mathbf{1}_{N_W}}{\sqrt{N_W}}\right|. \tag{12}$$

A high value of $\eta(\mathbf{Z}_i)$ suggests that the dominant key-side direction is broadly shared across key positions rather than concentrated on a few. As shown in the bottom row of Fig. 4, $\eta(\mathbf{Z}_i)$ remains high across timesteps, with medians around 0.84–0.85. These observations suggest that the dominant interaction between queries in W and keys in E_i is nearly rank-one and weakly dependent on individual key positions. This structure is consistent with local redundancy among keys inside the same external window.

Compensation-Aware Merging (CAM). In token merging, tokens within each local window are split into source and destination tokens, and each source token is merged into a matched nearby destination. Destination allocation thus determines the compression level of each region: a region with fewer destinations undergoes stronger feature aggregation, whereas one with more destinations better preserves local features. CAM allocates more destination tokens to regions where merging is likely to leave large residual errors even after CWA compensation, based on two observations.

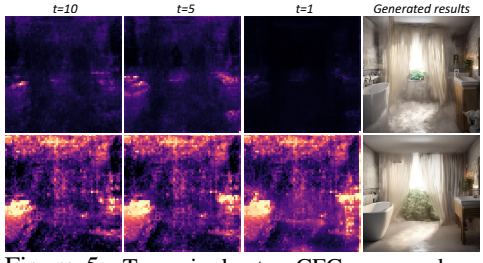


Figure 5: Top: single-step CFG maps, where high-response regions shift over time. Bottom: cumulative CFG maps retain previously important regions.

Table 1: Quantitative comparison with token merging methods for PixArt-α (Chen et al., 2024) on the MS-COCO 2017 validation set (Lin et al., 2014). Best results at each reduction ratio in bold.

	Method	FID↓	sFID↓	CLIP↑	FLOPs (T)	Δ	Latency (s)	Δ
-	PixArt-α (Chen et al., 2024)	37.62	38.61	31.75	260.37	-	9.67	-
r=0.3	ToMeSD (Bolya & Hoffman, 2023)	53.68	51.68	30.99	226.98	-12.8%	9.48	-2.0%
	ToMA (Lu et al., 2025b)	59.60	53.26	30.83	226.17	-13.1%	8.67	-10.3%
	Imp-ToMe (Wu et al., 2025)	38.95	46.07	31.45	226.98	-12.8%	9.54	-1.3%
	ComE-ToMe (ours)	**36.91**	**39.74**	**31.57**	231.58	-11.1%	9.51	-1.7%
r=0.5	ToMeSD (Bolya & Hoffman, 2023)	85.12	62.05	29.97	209.01	-19.7%	8.01	-17.2%
	ToMA (Lu et al., 2025b)	101.96	64.21	29.51	209.15	-19.7%	7.73	-20.1%
	Imp-ToMe (Wu et al., 2025)	45.72	56.18	31.03	209.01	-19.7%	8.08	-16.4%
	ComE-ToMe (ours)	**37.58**	**42.15**	**31.38**	213.61	-18.0%	8.00	-17.3%
r=0.7	ToMeSD (Bolya & Hoffman, 2023)	112.99	70.93	28.70	194.53	-25.3%	7.61	-21.3%
	ToMA (Lu et al., 2025b)	142.73	78.97	27.12	192.13	-26.2%	6.80	-29.7%
	Imp-ToMe (Wu et al., 2025)	64.27	71.69	30.15	194.53	-25.3%	7.67	-20.7%
	ComE-ToMe (ours)	**40.26**	**46.92**	**31.02**	199.12	-23.5%	7.57	-21.7%

First, tokens near window boundaries have part of their local context outside the compensation window, which makes $\Delta\hat{f}_W$ less accurate. We therefore assign more destination tokens near window boundaries so that these regions are merged less. Second, token importance varies over denoising timesteps. As shown in the top row of Fig. 5, a region with a high CFG response at $t = 10$ may respond weakly later, even though it may have already shaped local structures that aggressive merging can still distort. We thus maintain a cumulative CFG map, which retains previously important regions and provides a stable signal for allocation (bottom row of Fig. 5). Implementation details are provided in Appendix B.

4 Experiment

In this section, we describe implementation details (Sec. 4.1), and compare our ComE-ToMe with other token reduction methods (Sec. 4.2). We then provide a detailed analysis of our method (Sec. 4.3). More experimental results and analysis can be found in Appendix.

4.1 Implementation details

Datasets and models. We apply ComE-ToMe to various DiT architectures and perform extensive experiments on standard benchmarks for the text-to-image and text-to-video generation tasks. We use PixArt-α (Chen et al., 2024) and Stable Diffusion 3.5 medium (SD 3.5) (Esser et al., 2024) for text-to-image generation, and evaluate them on 5K samples on the validation set for the prompts of MS-COCO 2017 (Lin et al., 2014). We use DPM-Solver++ with the multistep scheduler (Lu et al., 2025a) with 20 steps for PixArt-α. For SD3.5, we use the flow-matching Euler scheduler (Esser et al., 2024) with 40 steps. For text-to-video generation, we use Wan2.1-1.3B (Wan et al., 2025), and evaluate them on the VBench benchmark (Huang et al., 2024b). We use the flow-prediction UniPC multistep scheduler (Zhao et al., 2023) with 50 steps. Following the implementation of Imp-ToMe (Wu et al., 2025), we apply CAM to attention modules within the middle half of the transformer layers across all denoising steps. Within this subset of merged layers, we apply our CWA framework to the deeper half of the transformer blocks.

Evaluation and metric. For the text-to-image generation task, we generate 5K images of MS-COCO 2017 prompts at 768×768 for PixArt-α and at 1024×1024 for SD 3.5. We evaluate the image quality using Fréchet Inception Distance (FID) (Heusel et al., 2017) and sFID (Nash et al., 2021), and we additionally use the CLIP score (Hessel et al., 2021) to measure the alignment with the text prompts. For text-to-video generation, we generate 946 videos at 480×640 resolution with 80 frames at FPS=16, corresponding to 5 second clips. We perform an evaluation using the overall VBench score, together with the Quality and Semantic scores. For all tasks, we report FLOPs and latency to evaluate computational efficiency.

4.2 Results

Quantitative results. We show in Tables 1, 2, and 3 quantitative comparisons of ComE-ToMe and state-of-the-art token merging methods (Bolya & Hoffman, 2023; Lu et al., 2025b; Wu et al., 2025) for text-to-image and text-to-video generation, respectively. We summarize our findings as follows: (1) ComE-ToMe outperforms existing token merging methods on all models and reduction ratios, with particularly large gains in sFID (Nash et al., 2021). This suggests that explicitly compensating for merging errors recovers the information lost during token merging, which prior methods leave unaddressed regardless of their token merging criteria. (2) ComE-ToMe achieves these gains with

Table 2: Quantitative comparison with token merging methods for SD3.5 (Esser et al., 2024) on the MS-COCO 2017 validation set (Lin et al., 2014). Best results at each reduction ratio in bold.

	Method	FID↓	sFID↓	CLIP↑	FLOPs (T)	Δ	Latency (s)	Δ
-	SD 3.5 (Esser et al., 2024)	27.95	48.50	32.15	899.90	-	48.46	-
r=0.3	ToMeSD (Bolya & Hoffman, 2023)	42.54	60.83	31.75	780.00	-13.3%	42.26	-12.8%
	ToMA (Lu et al., 2025b)	37.30	61.02	31.96	775.97	-13.8%	40.47	-16.5%
	Imp-ToMe (Wu et al., 2025)	27.23	51.31	32.11	780.00	-13.3%	42.21	-12.9%
	ComE-ToMe (ours)	**26.64**	**44.38**	**32.27**	799.29	-11.2%	42.86	-11.6%
r=0.5	ToMeSD (Bolya & Hoffman, 2023)	69.33	67.43	31.13	715.60	-20.5%	36.47	-24.7%
	ToMA (Lu et al., 2025b)	65.01	68.97	31.27	715.91	-20.4%	35.76	-26.2%
	Imp-ToMe (Wu et al., 2025)	31.49	56.64	32.05	715.60	-20.5%	36.47	-24.7%
	ComE-ToMe (ours)	**26.99**	**51.02**	**32.06**	734.90	-18.3%	36.81	-24.0%
r=0.7	ToMeSD (Bolya & Hoffman, 2023)	92.01	70.84	30.55	663.83	-26.2%	33.21	-31.5%
	ToMA (Lu et al., 2025b)	101.31	73.56	29.93	652.98	-27.4%	31.24	-35.5%
	Imp-ToMe (Wu et al., 2025)	45.80	65.61	31.73	663.83	-26.2%	33.26	-31.4%
	ComE-ToMe (ours)	**32.05**	**53.92**	**31.93**	683.12	-24.1%	32.89	-32.1%

Table 3: Quantitative comparison with token merging methods for Wan2.1 (Wan et al., 2025) on the VBench (Huang et al., 2024b). Best results at each reduction ratio in bold.

	Method	VBench↑	Semantic↑	Quality↑	FLOPs (T)	Δ	Latency (s)	Δ
-	Wan2.1-1.3B (Wan et al., 2025)	77.31	57.78	82.19	18260.91	-	173.56	-
r=0.3	ToMeSD (Bolya & Hoffman, 2023)	72.49	50.91	77.88	14696.66	-19.5%	144.83	-16.6%
	ToMA (Lu et al., 2025b)	64.31	33.65	71.98	15217.17	-16.7%	148.70	-14.3%
	Imp-ToMe (Wu et al., 2025)	72.64	52.09	77.77	14696.66	-19.5%	144.32	-16.8%
	ComE-ToMe (ours)	**74.95**	**56.51**	**79.56**	14762.39	-19.2%	144.54	-16.7%
r=0.5	ToMeSD (Bolya & Hoffman, 2023)	63.28	32.25	71.04	12944.74	-29.1%	128.62	-25.9%
	ToMA (Lu et al., 2025b)	59.78	24.15	68.68	12947.73	-29.1%	127.33	-26.6%
	Imp-ToMe (Wu et al., 2025)	66.82	40.59	73.38	12944.74	-29.1%	128.25	-26.1%
	ComE-ToMe (ours)	**70.48**	**47.01**	**76.35**	13010.48	-28.8%	128.71	-25.8%
r=0.7	ToMeSD (Bolya & Hoffman, 2023)	57.74	24.75	65.99	11692.24	-36.0%	116.97	-32.6%
	ToMA (Lu et al., 2025b)	57.77	21.99	66.72	11868.29	-35.0%	117.07	-32.5%
	Imp-ToMe (Wu et al., 2025)	61.69	29.50	69.73	11692.24	-36.0%	116.27	-33.0%
	ComE-ToMe (ours)	**62.79**	**31.64**	**70.58**	11757.97	-35.6%	116.66	-32.8%

marginal overhead. Compared with Imp-ToMe (Wu et al., 2025), ComE-ToMe increases FLOPs by at most 2.9% and its latency remains within 1.6% across all settings. This allows ComE-ToMe to adopt more aggressive reduction ratios while achieving better generation quality. For example, on PixArt-α, ComE-ToMe with $r = 0.7$ achieves lower FID and sFID than Imp-ToMe with $r = 0.5$ (40.26 vs. 45.72 and 46.92 vs. 56.18), while running faster (7.57s vs. 8.08s). We provide the quality-efficiency trade-off for all models in Fig. 10 in the Appendix.

Qualitative results. We show in Fig. 1 and Fig. 6 qualitative comparisons of generated images using PixArt-α (Chen et al., 2024) and SD 3.5 (Esser et al., 2024), respectively. We observe that ComE-ToMe produces more realistic images that are visually closer to those generated by the baseline models. For example, as shown in Fig. 6, ToMeSD (Bolya & Hoffman, 2023) and ToMA (Lu et al., 2025b) introduce visible artifacts on both the cow and the teddy bear. Imp-ToMe (Wu et al., 2025) preserves the overall layout but loses fine details of the cow and blurs the texture of the chair. In contrast, ComE-ToMe preserves both the foreground objects and the surrounding details, indicating that ComE-ToMe effectively reduces the merging error in the output features. We also show in Fig. 7 generated videos using Wan2.1-1.3B (Wan et al., 2025). We can see that ComE-ToMe produces high-fidelity videos that are similar to those from the full-token model, demonstrating the effectiveness of ComE-ToMe. Please see Appendix C for more results.

4.3 Discussions

Ablations. We show in Table 4 the ablation study for each component of ComE-ToMe at $r = 0.5$. We observe that CWA improves performance across different token merging methods, showing that compensating for merging errors is generally effective regardless of the merging strategy. For instance, applying CWA to ToMeSD reduces FID from 85.12 to 47.64. When combined with CAM, CWA further improves performance across all reduction ratios (see Table 9 in the appendix for full results). This indicates that CAM complements CWA by assigning more destination tokens to compensation-sensitive regions.

Table 4: Ablation study on the components of our methods.

CWA	Merging method	r=0.5		
		FID↓	sFID↓	CLIP↑
✗	ToMeSD (Bolya & Hoffman, 2023)	85.12	62.05	29.97
	ToMA (Lu et al., 2025b)	101.96	64.21	29.51
	Imp-ToMe (Wu et al., 2025)	**45.72**	56.18	**31.03**
	CAM	46.04	**54.72**	30.88
✓	ToMeSD (Bolya & Hoffman, 2023)	47.64	51.69	31.12
	ToMA (Lu et al., 2025b)	55.88	54.53	30.84
	Imp-ToMe (Wu et al., 2025)	38.36	49.49	31.33
	CAM	**37.58**	**42.15**	**31.38**

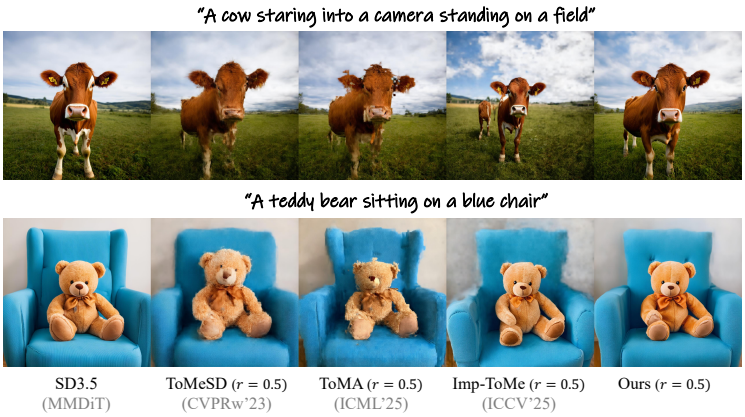


Figure 6: Qualitative comparisons of text-to-image generation for Stable Diffusion 3.5 (Esser et al., 2024) on MS-COCO 2017 validation set (Lin et al., 2014).

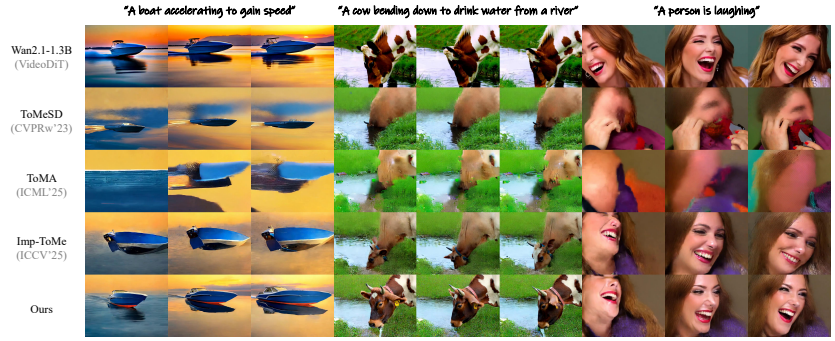


Figure 7: Qualitative comparisons of text-to-video generation for Wan2.1-1.3B (Wan et al., 2025) on VBench (Huang et al., 2024b).

Window size. We show in Fig. 8 the effect of the number of windows L. We report the relative FLOPs of self-attention layers and the corresponding FID on PixArt-α at $r = 0.5$, while varying L from 1 to 256. We can see that the two curves exhibit opposing trends with respect to L. When L is small, each window contains a larger number of tokens. As a result, CWA performs attention over increasingly large local regions, and its FLOPs approach those of full-token attention. When L is large, the windows shrink, and the FLOPs decrease. However, the number of tokens in the windows becomes too small to retain sufficient local context, which degrades the quality of the compensation. We therefore set $L = 16$ as the default, which achieves nearly the same FLOPs as merging-only while maintaining a low FID.

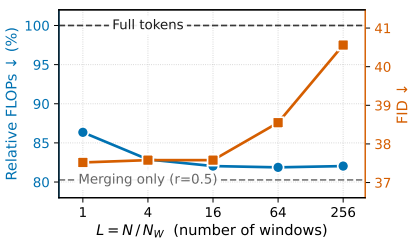


Figure 8: Trade-off curves of FLOPs and FID by varying L.

5 Conclusion

In this paper, we have addressed the detail degradation caused by token merging in diffusion transformers. We introduced ComE-ToMe, a training-free token merging framework that compensates for merging-induced feature distortion by exploiting the locality of attention. ComE-ToMe combines two components: CWA, which estimates and compensates for local merging errors through calibrated window attention, and CAM, which allocates destination tokens to regions where merging errors are likely to be more harmful. Experiments on standard image and video generation benchmarks show that ComE-ToMe improves generation quality over existing token merging methods while preserving acceleration benefits. Further analysis validates the role of locality and the effectiveness of each component.

AI USE STATEMENT

In this work, we used generative AI tools to improve the readability of the paper, including grammar checking and refining sentences. We have not used generative AI tools for developing methods or analyzing results. All AI-assisted work was reviewed by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

ComE-ToMe is a training-free acceleration framework for DiTs. By reducing the computational cost of inference without retraining, our work can lower the energy footprint and latency of large-scale image and video generation, making these models more accessible under limited compute budgets. As an inference acceleration technique, ComE-ToMe does not introduce new generative capabilities but inherits the behaviors of the pretrained models. Consequently, concerns associated with diffusion-based generation, such as the synthesis of misleading or unauthorized content, also apply to outputs produced under our framework. We encourage the use of standard safeguards such as content provenance and watermarking when ComE-ToMe is deployed in practice.

REPRODUCIBILITY STATEMENT

We describe CWA and CAM in Sec. 3.3, and provide the experimental settings, including the models, samplers, number of denoising steps, datasets, resolutions, and evaluation metrics, in Sec. 4.1. We provide further implementation details in Appendix B, including the efficient implementation of CWA, the hyperparameters of CAM, the adaptations of CWA to SD 3.5 and Wan2.1, and the libraries used to compute the metrics. The equivalence used for the batched implementation of CWA and the proof of Proposition 1 are presented in Appendix A.1 and A.2, respectively. We also specify the hardware used for all experiments in Appendix F. All models and benchmarks used in our experiments are publicly available, and we will make our code publicly available upon acceptance.

REFERENCES

Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *CVPRW*, 2023.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *ICLR*, 2023.

Junsong Chen, YU Jincheng, GE Chongjian, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *ICLR*, 2024.

Ziyi Dong, Chengxing Zhou, Weijian Deng, Pengxu Wei, Xiangyang Ji, and Liang Lin. Can we achieve efficient diffusion without self-attention? distilling self-attention into convolutions. In *ICCV*, 2025.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

Haipeng Fang, Sheng Tang, Juan Cao, Enshuo Zhang, Fan Tang, and Tong-Yee Lee. Attend to not attended: Structure-then-detail token merging for post-training dit acceleration. In *CVPR*, 2025.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPSW*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *NeurIPS*, 2022.

Yushi Huang, Ruihao Gong, Jing Liu, Tianlong Chen, and Xianglong Liu. Tfmq-dm: Temporal feature maintenance quantization for diffusion models. In *CVPR*, 2024a.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024b.

Dahye Kim, Deepti Ghadiyaram, and Raghudeep Gadde. Ddit: Dynamic patch scheduling for efficient diffusion transformers. In *CVPR*, 2026.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *ECCV*, 2022.

Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. In *ICLR*, 2023.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *NeurIPS*, 2024.

Dong-Jae Lee, Jiwan Hur, Jaehyun Choi, Jaemyung Yu, and Junmo Kim. Frequency-aware token reduction for efficient vision transformer. In *NeurIPS*, 2025a.

Sanghyeok Lee, Joonmyung Choi, and Hyunwoo J Kim. Multi-criteria token fusion with one-step-ahead attention for efficient vision transformers. In *CVPR*, 2024.

Seunghoon Lee, Jeongwoo Choi, Byunggwan Son, Jaehyeon Moon, Jeimin Jeon, and Bumsub Ham. Accuquant: Simulating multiple denoising steps for quantizing diffusion models. In *NeurIPS*, 2025b.

Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *ICCV*, 2023.

Youwei Liang, GE Chongjian, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Evit: Expediting vision transformers via token reorganizations. In *ICLR*, 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

Dongyang Liu, Meina Kan, Shiguang Shan, and Xilin Chen. A simple romance between multi-exit vision transformer and token reduction. In *ICLR*, 2024.

Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. In *CVPR*, 2025a.

Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Junjie Chen, and Linfeng Zhang. From reusing to forecasting: Accelerating diffusion models with taylorseers. In *ICCV*, 2025b.

Sifan Long, Zhen Zhao, Jimin Pi, Shengsheng Wang, and Jingdong Wang. Beyond attentive tokens: Incorporating token importance and diversity for efficient vision transformers. In *CVPR*, 2023.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 2025a.

Wenbo Lu, Shaoyi Zheng, Yuxuan Xia, and Shengjie Wang. Toma: Token merge with attention for diffusion models. In *ICML*, 2025b.

Yanzuo Lu, Manlin Zhang, Andy J Ma, Xiaohua Xie, and Jianhuang Lai. Coarse-to-fine latent diffusion for pose-guided person image synthesis. In *CVPR*, 2024.

Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. In *ICML*, 2021.

Bowen Pan, Rameswar Panda, Yifan Jiang, Zhangyang Wang, Rogerio Feris, and Aude Oliva. Ia-red^2: Interpretability-aware redundancy reduction for vision transformers. In *NeurIPS*, 2021.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *CVPR*, 2022.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *CVPR*, 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024.

Yurui Qian, Qi Cai, Yingwei Pan, Yehao Li, Ting Yao, Qibin Sun, and Tao Mei. Boosting diffusion models with moving average sampling in frequency domain. In *CVPR*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*, 2021.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *ACM SIGKDD*, 2020.

Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. In *ICLR*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022.

Pratheba Selvaraju, Tianyu Ding, Tianyi Chen, Ilya Zharkov, and Luming Liang. Fora: Fast-forward caching in diffusion transformer acceleration. *arXiv preprint arXiv:2407.01425*, 2024.

Byunggwan Son, Jeimin Jeon, Jeongwoo Choi, and Bumsub Ham. Relational feature caching for accelerating diffusion transformers. In *ICLR*, 2026.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024.

Chau Tran, Duy MH Nguyen, Manh-Duy Nguyen, TrungTin Nguyen, Ngan Le, Pengtao Xie, Daniel Sonntag, James Y Zou, Binh Nguyen, and Mathias Niepert. Accelerating transformers with spectrum-preserving token merging. In *NeurIPS*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Hongjie Wang, Difan Liu, Yan Kang, Yijun Li, Zhe Lin, Niraj K Jha, and Yuchen Liu. Attention-driven training-free efficiency enhancement of diffusion models. In *CVPR*, 2024.

Siyuan Wei, Tianzhu Ye, Shen Zhang, Yao Tang, and Jiajun Liang. Joint token pruning and squeezing towards more aggressive compression of vision transformers. In *CVPR*, 2023.

Haoyu Wu, Jingyi Xu, Hieu Le, and Dimitris Samaras. Importance-based token merging for efficient image and video generation. In *ICCV*, 2025.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. In *ICLR*, 2025.

Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. Evo-vit: Slow-fast token evolution for dynamic vision transformer. In *AAAI*, 2022.

Xinye Yang, Yuxin Yang, Haoran Pang, Aaron Xuxiang Tian, and Luking Li. Freqts: Frequency-aware token selection for accelerating diffusion models. In *AAAI*, 2025a.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *ICLR*, 2025b.

Yuzhe Yao, Feng Tian, Jun Chen, Haonan Lin, Guang Dai, Yong Liu, and Jingdong Wang. Timestep-aware correction for quantized diffusion models. In *ECCV*, 2024.

Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *CVPR*, 2022.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. In *NeurIPS*, 2023.

APPENDIX

In this appendix, we provide the following contents:

- Theoretical analyses of CWA, including its batched implementation and the optimality of the averaged key (Appendix A).
- Implementation details of CWA and CAM (Appendix B).
- Additional quantitative and qualitative results (Appendix C).
- Ablation studies and further analyses of the merging error (Appendix D).
- Limitations of our method (Appendix E).
- Hardware used in our experiments (Appendix F).

A THEORETICAL ANALYSIS

A.1 REDUCING CWA LATENCY VIA BATCHED IMPLEMENTATION

In this section, we show that proportional attention (Bolya et al., 2023) on merged tokens $\bar{\mathbf{X}}$ is equivalent to standard attention on the unmerged sequence whose token features are duplicated, *i.e.*, $\mathbf{U}\bar{\mathbf{X}}$, according to the merge assignments.

Let $\bar{\mathbf{X}} \in \mathbb{R}^{N' \times d}$ be the merged token features, where $N' < N$. Let $\mathbf{U} \in \{0,1\}^{N \times N'}$ be the unmerging matrix. Each row of $\mathbf{U}$ has exactly one nonzero entry, so that each original token position is assigned to one merged token. We write $g(i)$ for the merged token index assigned to original position i, *i.e.*,

$$U_{i,g(i)} = 1.$$

Then the unmerged feature at position i is

$$(\mathbf{U}\bar{\mathbf{X}})_i = \bar{\mathbf{x}}_{g(i)}.$$

Let

$$s_j = \sum_{i=1}^{N} U_{ij}$$

denote the number of original tokens represented by merged token j. We first consider standard attention computed on the unmerged sequence $\mathbf{U}\bar{\mathbf{X}}$. For notational simplicity, we write

$$\bar{\mathbf{q}}_j = W_Q \bar{\mathbf{x}}_j, \qquad \bar{\mathbf{k}}_j = W_K \bar{\mathbf{x}}_j, \qquad \bar{\mathbf{v}}_j = W_V \bar{\mathbf{x}}_j.$$

For an original token position i, the corresponding query is $\bar{\mathbf{q}}_{g(i)}$. Standard attention on $\mathbf{U}\bar{\mathbf{X}}$ gives

$$f(\mathbf{U}\bar{\mathbf{X}})_i = \sum_{\ell=1}^{N} \frac{\exp\left(\bar{\mathbf{q}}_{g(i)}^\top \bar{\mathbf{k}}_{g(\ell)}/\sqrt{d}\right)}{\sum_{r=1}^{N} \exp\left(\bar{\mathbf{q}}_{g(i)}^\top \bar{\mathbf{k}}_{g(r)}/\sqrt{d}\right)} \bar{\mathbf{v}}_{g(\ell)}. \tag{13}$$

Since all original positions assigned to the same merged token share the same key and value, we can group the summation by merged token index. For each merged token j, there are s_j original positions whose key-value pair is $(\bar{\mathbf{k}}_j, \bar{\mathbf{v}}_j)$. Therefore,

$$f(\mathbf{U}\bar{\mathbf{X}})_i = \sum_{j=1}^{N'} \frac{s_j \exp\left(\bar{\mathbf{q}}_{g(i)}^\top \bar{\mathbf{k}}_j/\sqrt{d}\right)}{\sum_{m=1}^{N'} s_m \exp\left(\bar{\mathbf{q}}_{g(i)}^\top \bar{\mathbf{k}}_m/\sqrt{d}\right)} \bar{\mathbf{v}}_j. \tag{14}$$

Now we consider attention computed directly on the merged sequence $\bar{\mathbf{X}}$. In proportional attention, the attention logit for key-value token j is biased by $\log s_j$. Thus,

$$f_{\text{prop}}(\bar{\mathbf{X}})_a = \sum_{j=1}^{N'} \frac{\exp\left(\bar{\mathbf{q}}_a^\top \bar{\mathbf{k}}_j/\sqrt{d} + \log s_j\right)}{\sum_{m=1}^{N'} \exp\left(\bar{\mathbf{q}}_a^\top \bar{\mathbf{k}}_m/\sqrt{d} + \log s_m\right)} \bar{\mathbf{v}}_j. \tag{15}$$

Using $\exp(z + \log s_j) = s_j \exp(z)$, we obtain

$$f_{\text{prop}}(\bar{\mathbf{X}})_a = \sum_{j=1}^{N'} \frac{s_j \exp\left(\bar{\mathbf{q}}_a^\top \bar{\mathbf{k}}_j / \sqrt{d}\right)}{\sum_{m=1}^{N'} s_m \exp\left(\bar{\mathbf{q}}_a^\top \bar{\mathbf{k}}_m / \sqrt{d}\right)} \bar{\mathbf{v}}_j. \tag{16}$$

By setting $a = g(i)$, Eq. (16) becomes identical to Eq. (14). Hence, for every original token position i,

$$f(\mathbf{U}\bar{\mathbf{X}})_i = f_{\text{prop}}(\bar{\mathbf{X}})_{g(i)}. \tag{17}$$

This is exactly the i-th row of $\mathbf{U} f_{\text{prop}}(\bar{\mathbf{X}})$. Therefore,

$$\boxed{f(\mathbf{U}\bar{\mathbf{X}}) = \mathbf{U} f_{\text{prop}}(\bar{\mathbf{X}})}. \tag{18}$$

This equivalence directly simplifies the CWA implementation. By the result above, the term $\mathbf{U} f_W(\bar{\mathbf{X}})$ in Eq. (2) can be equivalently written as $f_W(\mathbf{U}\bar{\mathbf{X}})$.

In practice, we implement CWA by computing $\tilde{f}_W(\mathbf{U}\bar{\mathbf{X}})$ as an alternative of the term $\mathbf{U} f_W(\bar{\mathbf{X}})$. The term $\mathbf{U}\tilde{f}_W(\bar{\mathbf{X}})$ yields the same output by the equivalence above, but differs in implementation. Fig. 3 depicts the correction in the form $\mathbf{U}\tilde{f}_W(\bar{\mathbf{X}})$ for consistency with the merge-compute-unmerge pipeline. However, directly computing this form requires applying $\tilde{f}_W$ to each local window separately and then unmerging the outputs. Since the number of merged tokens varies across windows, batching is infeasible and an explicit loop over windows is required. Instead, we implement the equivalent form $\tilde{f}_W(\mathbf{U}\bar{\mathbf{X}})$, which first restores the full-resolution token layout and then computes $\tilde{f}_W$ over all local windows in a single batched operation, reducing the latency overhead of CWA.

A.2 Optimality of the Averaged Key Approximation

We analyze why the averaged key in Eq. (5) provides a reasonable proxy for the external-window keys. Consider a logit matrix $\mathbf{Z}_i \in \mathbb{R}^{N_W \times N_W}$ between the queries in W and the keys in an external window E_i, whose (m, n)-th element is defined as

$$z_{mn} = \frac{1}{\sqrt{d}} \mathbf{q}_m^\top \mathbf{k}_n, \tag{19}$$

where we denote by $\mathbf{q}_m$ and $\mathbf{k}_n$ the m-th query in W and the n-th key in E_i, respectively. We replace the keys in E_i with a single averaged key $\bar{\mathbf{k}}_i$, yielding identical logits across all key positions for each query. We show that using the averaged key $\bar{\mathbf{k}}_i$ in Eq. (5) minimizes the discrepancy between the original and approximated logit matrices, as follows:

Proposition 1. *The averaged key $\bar{\mathbf{k}}_i$ in Eq. (5) minimizes the discrepancy between the original logit matrix $\mathbf{Z}_i \in \mathbb{R}^{N_W \times N_W}$ and its averaged-key approximation as follows:*

$$\bar{\mathbf{k}}_i = \arg\min_{\mathbf{k}} \|\mathbf{Z}_i - \frac{1}{\sqrt{d}} \mathbf{Q}\mathbf{k}\mathbf{1}_{N_W}^\top\|_F^2 \tag{20}$$

where we denote by $\mathbf{1}_{N_W} \in \mathbb{R}^{N_W}$ and $\mathbf{Q} \in \mathbb{R}^{N_W \times d}$ a vector consisting of N_W ones and the matrix whose rows correspond to the queries in the local window, respectively.

Proof. The objective decomposes per row, $\|\mathbf{Z} - \mathbf{s}\,\mathbf{1}_m^\top\|_F^2 = \sum_{a=1}^{n_q} \sum_{b=1}^{m} (Z_{ab} - s_a)^2$, and each row is uniquely minimized by its mean $s_a^\star = m^{-1} \sum_{b=1}^{m} Z_{ab}$. □

Proposition 1 shows that the averaged key $\bar{\mathbf{k}}_i$ is optimal as a single-key representative for the external window. The resulting approximated logit matrix has identical columns, thereby representing a rank-one pattern. As shown above, the original logit matrix $\mathbf{Z}_i$ is well approximated by a rank-one structure. Under this observation, we seek the best rank-one proxy of the form $\mathbf{s}\mathbf{1}^\top$, where all columns share the same representative key. The Frobenius-norm minimizer of $\|\mathbf{Z}_i - \mathbf{s}\mathbf{1}^\top\|_F^2$ is obtained by setting $\mathbf{s}$ to the row mean of $\mathbf{Z}_i$. Since $\mathbf{Z}_i = \mathbf{Q}_W \mathbf{K}_{E_i}^\top / \sqrt{d}$, this row mean corresponds to using the averaged key of the external window.

B Implementation Details

B.1 Efficient CWA Implementation

Instead of applying CWA separately to the conditional and unconditional branches of classifier-free guidance (CFG) (Ho & Salimans, 2021), which nearly doubles its cost, we average the two branch features and run CWA only once. Since both branches are computed from the same noisy latent and differ only in the conditioning input, they share the same token layout and similar local query-key patterns; their feature maps indeed show an average per-token cosine similarity close to one (Fig. 9). The averaged compensation thus closely approximates the branch-specific one at half the cost. To further alleviate latency, we compute the merging matrix only at the first merged block of each denoising step and reuse it across all subsequent blocks within the same step.

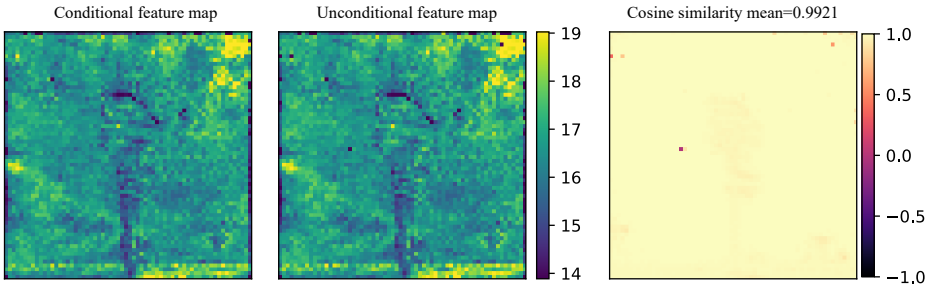


Figure 9: Similarity between conditional and unconditional feature maps.

B.2 Details of Compensation-Aware Merging

We implement CAM with two allocation signals: temporal CFG importance and boundary-aware preservation. Let $C_{t,i}$ be the normalized CFG score of token i at timestep t. Since early denoising steps mainly determine coarse structure, we start accumulating CFG scores only from the middle of the denoising trajectory. This prevents early coarse structural responses from being permanently retained in the cumulative map. After the accumulation starts, we update the cumulative CFG map as

$$S_{t,i} = \max\left(\alpha S_{t-1,i}, C_{t,i}\right), \tag{21}$$

with $\alpha = 0.9$ in all experiments. This update selects the larger value between the current CFG score and the decayed cumulative score, allowing previously important regions to remain protected when their current CFG response decreases.

We then combine this temporal signal with a boundary-aware preservation term. Let $B_i \in \{0, 1\}$ indicate whether token i lies near a compensation-window boundary. The final allocation score is

$$A_{t,i} = S_{t,i} + \gamma B_i, \tag{22}$$

where γ controls the strength of boundary preservation. We set $\gamma = 0.25$ in all experiments and use A_t for destination token allocation.

Table 5 analyzes the effect of the temporal accumulation factor α and the boundary preservation weight γ in CAM. Without either term, the method obtains an FID of 42.40, sFID of 52.46, and CLIP score of 30.88. Adding only the boundary term ($\gamma = 0.25$) improves FID to 40.72 and CLIP to 31.02, showing that preserving boundary tokens helps reduce artifacts near compensation-window boundaries, although it slightly increases sFID to 53.94. Using only temporal accumulation ($\alpha = 0.9$) improves all metrics and yields the lowest sFID of 46.51, indicating that the cumulative CFG map provides a more stable allocation signal across denoising steps.

When both terms are used, CAM achieves the best FID of 40.26 and the highest CLIP score of 31.02, while maintaining an sFID of 46.92 comparable to the accumulation-only setting. This shows that temporal accumulation and boundary preservation play complementary roles. The cumulative CFG map stabilizes destination-token allocation over time, while the boundary term protects regions where the CWA estimate is less reliable. We therefore use $\alpha = 0.9$ and $\gamma = 0.25$ as the default setting.

Table 5: Ablation study on α and γ in CAM on PixArt-α with $r = 0.7$.

α	γ	FID↓	sFID↓	CLIP↑
0	0	42.40	52.46	30.88
	0.25	40.72	53.94	**31.02**
0.9	0	41.30	**46.51**	30.90
	0.25	**40.26**	46.92	**31.02**

B.3 Architecture-Specific Adaptations of CWA

B.3.1 Stable Diffusion 3.5

Stable Diffusion 3.5 uses MMDiT blocks, where image tokens and text tokens are processed together, which is called JointAttention. When applying CWA to JointAttention, the external tokens for an image window are not limited to image tokens outside the window. They also include the text tokens that participate in the same JointAttention. For each image window W, we construct the calibrated key set as follows. We first divide the image feature tokens into local windows and represent each external image window by its averaged key. We then concatenate these external image-window proxies with all text-token keys. This gives the calibrated external key set used to compute $\hat{f}_W$. In other words, $\hat{f}_W$ is estimated using both the averaged image-window proxies and the full text-token keys, so that CWA remains consistent with the JointAttention structure of MMDiT.

B.3.2 Wan2.1

Wan2.1 is a video diffusion transformer that operates on a $T \times H_p \times W_p$ grid of latent tokens with 3D axial rotary position embedding (RoPE) (Su et al., 2024). Each block contains a self-attention layer over the visual tokens, a cross-attention layer that mixes visual queries with text-conditioned key-value tokens, and a feed-forward layer. Only the self-attention layer applies RoPE over the visual grid, so we apply CWA exclusively to this path. Specifically, we modify CWA for Wan2.1's 3D RoPE structure with two adaptations as follows:

(1) We partition each frame into spatial windows of size $w_h \times w_w$ that do not overlap, giving a total of $L = T(H_p/w_h)(W_p/w_w)$ local windows across the input token $\mathbf{X}$. For a window W, the external set E_i contains every other 3D window in the video, which includes both spatial neighbors in the same frame and windows at the same spatial position in other frames. We represent each external window by a single averaged key, following Eq. (5).

(2) 3D RoPE rotates each query and key by a phase that depends on its (t, h, w) position. If we average keys after RoPE, the rotated phases cancel across the N_W positions in the window, and the proxy fails to represent the content. Instead, we average the keys before RoPE and rotate the result by the RoPE phase at the geometric center of E_i. The proxy behaves as a single virtual token at the center of E_i.

B.4 Evaluation Metrics

We use clean-fid (Parmar et al., 2022) to measure FID, and the deepspeed library (Rasley et al., 2020) to calculate FLOPs. We also use ViT-B/16 models of CLIP (Radford et al., 2021) to calculate CLIP score.

C Additional Experimental Results

C.1 Comparison with SDTM

We evaluate both SDTM and ComE-ToMe on Stable Diffusion 3.5, where the official setting of SDTM applies directly. We also tried to apply SDTM to PixArt-α, but we could not faithfully reproduce its performance since the official implementation of SDTM supports only SD3-family models. As shown in Table 6, ComE-ToMe consistently outperforms SDTM in FID, sFID, and CLIP score at all reduction ratios. We attribute this to the fact that SDTM, despite its advanced structure-then-detail selection, still relies on the merge-compute-unmerge pipeline and thus inherits its merging error, whereas ComE-ToMe compensates for this error.

Table 6: Comparison with SDTM (Fang et al., 2025) on Stable Diffusion 3.5.

	Method	FID↓	sFID↓	CLIP↑	Latency (s)↓
$r = 0.3$	SDTM (Fang et al., 2025)	31.84	52.91	31.88	43.52
	ComE-ToMe (ours)	**26.64**	**44.38**	**32.27**	**42.86**
$r = 0.5$	SDTM (Fang et al., 2025)	37.07	56.72	31.54	36.95
	ComE-ToMe (ours)	**26.99**	**51.02**	**32.06**	**36.81**
$r = 0.7$	SDTM (Fang et al., 2025)	49.42	57.72	31.46	33.23
	ComE-ToMe (ours)	**32.05**	**53.92**	**31.93**	**32.89**

C.2 Comparison with FreqTS

Prior works have explored frequency-aware token merging for diffusion models, and we compare ComE-ToMe with FreqTS (Yang et al., 2025a), the most recent method in this line of research. Since the official implementation of FreqTS is not publicly available, we reimplemented it following the descriptions in the original paper. Our reimplementation provides stable results only at a reduction ratio of $r = 0.3$, and thus we evaluate both methods. As shown in Table 7, ComE-ToMe outperforms FreqTS in FID, sFID, and CLIP score, suggesting that compensating for the merging error is more effective than relying on frequency-aware token selection alone.

Table 7: Comparison with FreqTS (Yang et al., 2025a) on COCO-5k at $r = 0.3$.

Method	FID↓	sFID↓	CLIP↑
FreqTS (Yang et al., 2025a)	49.01	58.91	30.85
ComE-ToMe (ours)	**36.91**	**39.74**	**31.57**

C.3 Additional Quantitative Results

Results on SDXL. We also evaluate ComE-ToMe on SDXL, as reported in Table 8. Compared with Imp-ToMe, ComE-ToMe achieves slightly better FID and comparable CLIP scores across reduction ratios, while maintaining similar latency. The improvement is more marginal than the gain observed on PixArt-α. We attribute this difference to the architecture of SDXL. Unlike DiT-based models, SDXL combines convolutional layers with transformer blocks. The convolutional layers provide an additional local inductive bias and help preserve spatial details even when token merging is applied in the transformer blocks. As a result, the quality degradation caused by token merging is less severe in SDXL, leaving less room for compensation to improve the final generation quality. Nevertheless, ComE-ToMe consistently improves or matches Imp-ToMe while introducing negligible latency overhead.

Table 8: Quantitative comparison with Imp-ToMe (Wu et al., 2025) on MS-COCO 2014 validation 30K using SDXL (Podell et al., 2024).

	Method	FID↓	CLIP↑	Latency (s)↓
$r = 0.3$	Imp-ToMe (Wu et al., 2025)	19.78	31.92	7.63
	ComE-ToMe (ours)	**19.16**	**31.94**	7.63
$r = 0.5$	Imp-ToMe (Wu et al., 2025)	21.43	31.90	6.98
	ComE-ToMe (ours)	**20.37**	**31.90**	6.98
$r = 0.7$	Imp-ToMe (Wu et al., 2025)	26.22	31.63	6.86
	ComE-ToMe (ours)	**23.35**	**31.73**	6.85

Trade-off between quality and efficiency. We show in Fig. 10 the generation quality with respect to FLOPs and latency for all models used in this paper (*i.e.*, PixArt-α (Chen et al., 2024), SD3.5 (Esser et al., 2024), and Wan2.1 (Wan et al., 2025)). We can see that our approach provides better trade-offs than other token merging methods for all models, achieving better generation quality at comparable FLOPs and latency. For text-to-image generation, the quality of other methods drops sharply as the reduction ratio increases, while that of ours remains close to the full-token baseline. For example, ours with $r = 0.7$ on PixArt-α increases FID by only 2.64 over the baseline, whereas Imp-ToMe Wu et al. (2025) increases it by 26.65. This suggests that compensating for merging errors becomes more beneficial as merging becomes more aggressive. For text-to-video generation, ours also provides better trade-offs at all reduction ratios.

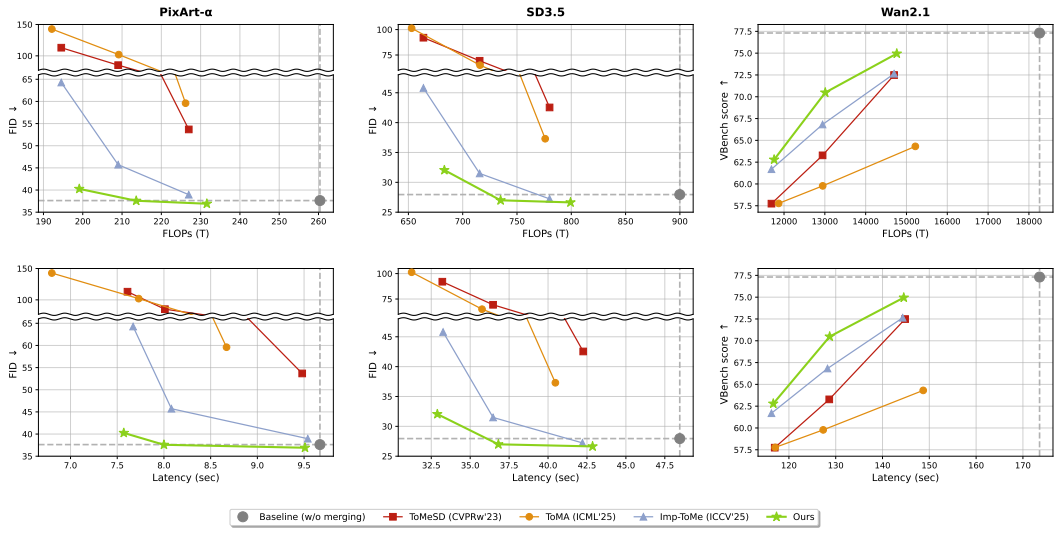


Figure 10: Comparison of trade-offs between generation quality and efficiency for PixArt-α (Chen et al., 2024), SD3.5 (Esser et al., 2024), and Wan2.1 (Wan et al., 2025). (Top) Quality-FLOPs. (Bottom) Quality-Latency.

C.4 Additional Qualitative Results

We show in Figs. 13, 14, and 15 additional qualitative comparisons for text-to-image and text-to-video generations, respectively. Our ComE-ToMe generates high-fidelity visual outputs consistently, whereas previous methods often introduce noticeable artifacts or blurred details compared to the baseline.

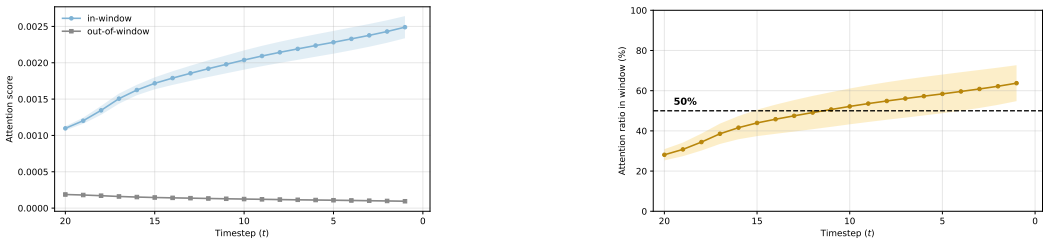
D Ablation Studies and Further Analysis

D.1 Component-wise Ablation

Table 9: Ablation study on the components of our methods.

CWA	Merging method	r=0.3			r=0.5			r=0.7		
		FID↓	sFID↓	CLIP↑	FID↓	sFID↓	CLIP↑	FID↓	sFID↓	CLIP↑
✗	ToMeSD (Bolya & Hoffman, 2023)	53.68	51.68	30.99	85.12	62.05	29.97	112.99	70.93	28.70
	ToMA (Lu et al., 2025b)	59.60	53.26	30.83	101.96	64.21	29.51	142.73	78.97	27.12
	Imp-ToMe (Wu et al., 2025)	38.95	46.07	**31.45**	**45.72**	56.18	**31.03**	**64.27**	71.69	**30.15**
	CAM	**38.60**	**45.80**	31.38	46.04	**54.72**	30.88	65.27	**66.46**	30.07
✓	ToMeSD (Bolya & Hoffman, 2023)	38.80	45.04	31.46	47.64	51.69	31.12	57.57	57.21	30.59
	ToMA (Lu et al., 2025b)	42.26	47.26	31.30	55.88	54.53	30.84	79.05	66.90	29.32
	Imp-ToMe (Wu et al., 2025)	36.93	42.55	31.55	38.36	49.49	31.33	42.40	52.46	30.88
	CAM	**36.91**	**39.74**	**31.57**	**37.58**	**42.15**	**31.38**	**40.26**	**46.92**	**31.02**

We show in Table 9 the full ablation study for each component of ComE-ToMe across reduction ratios $r = 0.3$, 0.5, and 0.7. A key observation is that CWA serves as a plug-and-play module that is applicable to any token merging method. Since CWA only requires the merging and unmerging matrices, $\mathbf{M}$ and $\mathbf{U}$, it is agnostic to how tokens are selected and merged. Indeed, applying CWA consistently improves FID, sFID, and CLIP score for all merging methods at all reduction ratios, including methods based on similarity-based merging (ToMeSD), linear-transform-based merging (ToMA), and importance-based merging (Imp-ToMe). The gain becomes larger as merging becomes more aggressive. For instance, CWA reduces FID of ToMeSD by 14.88, 37.48, and 55.42 at $r = 0.3$, 0.5, and 0.7, respectively, and that of ToMA from 142.73 to 79.05 at $r = 0.7$. Notably, with CWA, even the simple merging strategy of ToMeSD outperforms Imp-ToMe without compensation at $r = 0.3$ (38.80 vs. 38.95 FID) and $r = 0.7$ (57.57 vs. 64.27 FID). This suggests that compensating for merging errors is orthogonal to designing advanced token selection criteria, and can be even more effective.



(a) Per-token attention score.

(b) In-window attention ratio.

Figure 11: Attention locality in PixArt-α (Chen et al., 2024) on the MS-COCO 2017 validation set (Lin et al., 2014), measured at $L = 16$ ($N_W = 256$, $N = 4096$) across denoising timesteps.

Among the merging strategies, CAM benefits the most from CWA, particularly under aggressive merging. At $r = 0.7$, CWA reduces FID by 25.01 with CAM ($65.27 \rightarrow 40.26$), compared to 21.87 with Imp-ToMe ($64.27 \rightarrow 42.40$). As a result, combining CWA with CAM achieves the best FID, sFID, and CLIP scores across all reduction ratios. This indicates that CAM complements CWA by preserving tokens in regions where compensation errors are more likely to occur.

D.2 Attention Locality in Local Windows

We empirically verify the locality assumption that motivates CWA, where a small spatial neighborhood already carries most of the attention mass needed to estimate the merging error. Following the same protocol as Fig. 4, we measure attention statistics on PixArt-α (Chen et al., 2024) using the MS-COCO 2017 validation set (Lin et al., 2014), evaluated on all merging blocks (blocks 8–21 out of 28) and reported across denoising timesteps. The window size is set to our default $L = 16$ ($N_W = 256$ and $N = 4096$). Fig. 11(a) shows the average per-token attention score for keys inside the local window and for keys outside the window. The two curves differ by more than an order of magnitude across all timesteps, and the gap further widens at later timesteps as the in-window score grows while the out-of-window score continues to decay. Fig. 11(b) shows the cumulative attention mass that each query assigns to its local window. Although the in-window keys account for only $256/4096 = 6.25\%$ of all tokens, their share of the attention mass exceeds 50% in later timesteps. These observations are consistent with the high-frequency degradation at later timesteps in Fig. 2(b), and support compensating for merging errors within local windows.

D.3 Importance of Calibration in Window Attention

We compare Naive Window Attention (NWA), which applies window attention without any calibration, and our Calibrated Window Attention (CWA) both qualitatively (Fig. 12) and quantitatively (Table 10). NWA estimates the merging error using only tokens inside each local window. Since outside-window tokens are removed from the softmax denominator, the attention mass assigned to local tokens is overestimated. As shown in Fig. 12, this normalization mismatch produces distorted textures and visible artifacts. In contrast, CWA estimates the missing outside-window contribution and restores a more faithful softmax normalization, yielding natural textures without such artifacts. Consistently, CWA substantially outperforms NWA across all metrics in Table 10. This shows that calibration is crucial when local window attention is used to approximate the merging error.



Figure 12: Effect of calibration in window attention.

Table 10: Comparison of window attention variants on MS-COCO 2017 (Lin et al., 2014).

Attention	FID↓	sFID↓	CLIP↑
NWA	60.06	55.14	30.18
CWA (Ours)	**40.26**	**46.92**	**31.02**

D.4 Direct Validation of the Merging Error

We directly validate the estimated local merging error $\Delta\hat{f}_W$ (Eq. 8). Following $f = f_W + f_E$, the full merging error decomposes into local and out of window terms, $\Delta f = \Delta f_W + \Delta f_E$. We thus compare two pairs: $(\Delta f_W, \Delta f)$ to examine how much of the full error is local, and $(\Delta\hat{f}_W, \Delta f_W)$ to evaluate the averaged key calibration. We measure the errors in one of the CWA applied blocks of PixArt-α with $r = 0.7$ across denoising timesteps. Table 11 reports the cosine similarity and the norm ratio for each pair.

First, the comparison between Δf_W and Δf examines how much of the full merging error is captured by the local error. The local error Δf_W achieves a cosine similarity above 0.90 and a norm ratio of 1.00–1.02 with Δf at all timesteps, indicating that the local error determines the dominant direction and magnitude of the full merging error. This result can be understood through the frequency analysis in Sec. 3.2. Since merging acts as a low-pass filter, it mainly affects high-frequency components, while low-frequency components are well preserved in $\mathbf{U}f(\bar{\mathbf{X}})$ (Fig. 2(a)). The out-of-window contribution aggregates interactions with distant tokens and thus mainly carries coarse, low-frequency information, which is barely altered by merging. In contrast, the in-window interactions that form fine-grained details are directly affected by merging, so the merging error concentrates on the local term. These results support the design of ComE-ToMe, where $\mathbf{U}f(\bar{\mathbf{X}})$ provides long-range dependencies, while CWA focuses on the local error that dominates the merging error.

Second, the comparison between $\Delta\hat{f}_W$ and Δf_W evaluates the accuracy of the calibration in CWA. The estimate achieves a cosine similarity above 0.91 and a norm ratio of 0.965–0.998 with Δf_W at all timesteps, showing that the averaged-key proxy (Eq. 5) accurately recovers the softmax normalization over all tokens.

Table 11: Direct comparison of the merging errors on PixArt-α with $r = 0.7$. We report the mean and standard deviation over samples.

Metric	$t = 10$	$t = 8$	$t = 6$	$t = 4$	$t = 2$	$t = 0$
Locality: Δf_W vs. Δf						
Cosine ↑	0.910±0.017	0.909±0.021	0.905±0.023	0.913±0.022	0.913±0.024	0.913±0.028
Norm ratio → 1	1.022±0.017	1.021±0.019	1.017±0.019	1.008±0.019	1.004±0.019	1.000±0.018
Calibration: $\Delta\hat{f}_W$ vs. Δf_W						
Cosine ↑	0.923±0.016	0.920±0.021	0.916±0.023	0.916±0.026	0.916±0.026	0.917±0.030
Norm ratio → 1	0.965±0.015	0.966±0.017	0.970±0.017	0.989±0.018	0.996±0.018	0.998±0.019

D.5 Frequency Spectrum Analysis

Frequency spectrum of the merging error. In Sec. 3.2, we analyze the relative spectrum power of the unmerged output $\mathbf{U}f(\bar{\mathbf{X}})$. To directly examine which frequency components constitute the merging error, we further analyze the spectrum of the merging error Δf itself. Specifically, we measure the relative error spectrum, $100 \cdot |\mathcal{F}(\Delta f)|_{\text{band}} / |\mathcal{F}(f(\mathbf{X}))|_{\text{band}}$, where $\mathcal{F}(\cdot)$ denotes the 2D Fourier transform and $|\cdot|_{\text{band}}$ denotes the spectral magnitude within a frequency band. This measures the size of the merging error relative to the full-attention output within the same band. The low and high bands follow the definition in Fig. 2 ($0 \le \omega \le \frac{\pi}{4}$ and $\frac{3\pi}{4} \le \omega \le \pi$). We use the same setup as Fig. 2 and report the results for early ($t \in [11, 20]$) and late ($t \in [1, 10]$) timesteps in Table 12. For all merging methods, the relative error in the high band is 3.0–5.5$\times$ larger than that in the low band at early timesteps, and 1.8–3.7$\times$ larger at late timesteps. This confirms that the merging error Δf is dominated by high-frequency components, supporting our motivation to recover fine-grained details.

Table 12: Relative error spectrum (%) of the merging error Δf in the low and high frequency bands on PixArt-α (Chen et al., 2024) with $r = 0.5$.

Method	Early ($t \in [11, 20]$)			Late ($t \in [1, 10]$)		
	Low	High	High/Low	Low	High	High/Low
ToMeSD	15.8±1.4	86.8±1.3	5.5×	24.5±1.9	90.2±1.8	3.7×
ToMA	15.0±1.7	75.5±1.7	5.0×	26.3±3.1	71.8±2.8	2.7×
Imp-ToMe	24.8±2.6	74.2±2.1	3.0×	35.6±2.6	64.6±6.7	1.8×
Ours (merging only)	21.5±1.9	72.3±2.0	3.4×	34.0±2.5	63.5±4.4	1.9×

Effect of CWA on the frequency spectrum. We further examine whether CWA indeed recovers the high-frequency components lost by token merging. Table 13 reports the RSP of the output before and after applying CWA at the timesteps where CWA is applied, under the same setup as Fig. 2. At every timestep, CWA raises the high-frequency band RSP by $+5.6$ to $+8.9$ points, recovering roughly two thirds of the gap to the full-token baseline. In contrast, the low-frequency band RSP changes by at most $+1.7$ points. This shows that CWA selectively restores the high-frequency components while leaving the well-preserved low-frequency components nearly unchanged, which is consistent with our design of compensating for the local merging error.

Table 13: RSP (%) in the low and high frequency bands without and with CWA on PixArt-α (Chen et al., 2024).

t	Low band			High band		
	w/o CWA	w/ CWA	Δ	w/o CWA	w/ CWA	Δ
10	98.6±3.1	98.6±2.5	+0.0	91.9±5.8	97.5±4.7	+5.6
6	97.5±3.7	98.6±3.5	+1.1	88.4±7.8	95.7±4.7	+7.3
1	94.3±3.8	96.0±3.5	+1.7	88.0±5.1	96.9±3.5	+8.9

D.6 Compatibility with Feature Caching

Since ComE-ToMe reduces the cost within each denoising step, it can be combined with feature caching methods that skip computations across steps. We integrate ComE-ToMe with FORA (Selvaraju et al., 2024) ($N = 2$) and TeaCache (Liu et al., 2025a) ($\delta = 0.6$) on PixArt-α with $r = 0.7$. As shown in Table 14, caching further increases the speedup over the full-token model from $1.28\times$ to $1.44\times$ and $1.68\times$, respectively. Surprisingly, caching even improves generation quality (FID $40.26 \rightarrow 37.82$ with FORA), consistent with the observation in Imp-ToMe (Wu et al., 2025). We conjecture that, since the merging assignment changes across denoising steps, merging errors occur in different regions at each step, and reusing cached features from a previous step partially offsets the merging errors of the current step. We leave further investigation of this interplay for future work.

Table 14: Compatibility of ComE-ToMe with feature caching methods on PixArt-α with $r = 0.7$.

Method	FID↓	sFID↓	CLIP↑	Speedup↑
ComE-ToMe (ours)	40.26	46.92	31.02	1.28×
+ FORA ($N = 2$)	**37.82**	**43.90**	**31.34**	1.44×
+ TeaCache ($\delta = 0.6$)	39.37	47.13	31.14	**1.68**×

E Limitations

Our method has several limitations. First, CWA focuses on the local merging error and relies on $\mathbf{U}f(\bar{\mathbf{X}})$ for long-range dependencies. When interactions with distant tokens are critical but not faithfully modeled by $\mathbf{U}f(\bar{\mathbf{X}})$, the generation quality may degrade. Second, the averaged external-key proxy assumes that external-window logits can be represented by a compact window-level approximation. This assumption can be less accurate when the attention mass is concentrated on a few specific external tokens. Finally, CAM includes empirical design choices, such as the accumulation start timestep, the decay factor of the cumulative CFG map, and the boundary preservation weight.

We use fixed values across experiments, but more adaptive allocation rules may further improve robustness across models and sampling schedules.

F HARDWARE RESOURCES

For text-to-image generation, we run all experiments on eight NVIDIA RTX A5000 GPUs paired with an Intel Xeon Gold 6226R CPU. For text-to-video generation, we use four NVIDIA PRO 6000 Max-Q edition GPUs paired with an Intel Xeon Gold 6526Y CPU. The same hardware is used for all baselines and our method to ensure a fair comparison of FLOPs and latency.



Figure 13: Qualitative comparisons of text-to-image generation for PixArt-α (Chen et al., 2024) on MS-COCO 2017 validation set (Lin et al., 2014).



Figure 14: Qualitative comparisons of text-to-image generation for SD 3.5 (Esser et al., 2024) on MS-COCO 2017 validation set (Lin et al., 2014).



Figure 15: Qualitative comparisons of text-to-video generation for Wan2.1-1.3B (Wan et al., 2025) on VBench (Huang et al., 2024b).