Attend to Not Attended: Structure-then-Detail Token Merging for Post-training DiT Acceleration

Supplementary Material

We organize our supplementary material as follows.

Implementation Details:

- Sec. A.1: Details of preliminary experiments.
- Sec. A.2: Details of the implementation of backbones, our methodologies, and SoTA methods.

Additional Analyses:

- Sec. B.1: Analysis of reduced MACs across timesteps.
- Sec. B.2: Analysis of computation overhead of SSM and IDM merging mechanisms.

Additional Experiments:

- Sec. C.1: Extended comparisons with baseline models.
- Sec. C.2: Ablation study on image size effects.
- Sec. C.3: More visualizations of generated images.

A. Implementation Details

A.1. Details of Preliminaries

In Sec. 3, we examine the evolution of the LL, HL, LH, and HH subbands in the denoising process, along with the location and degree evolution of feature redundancies. Below, we detail the preliminary experiments.

Feature Collection. Utilizing the baseline model of Stable Diffusion 3 Medium and a 50-step Rectified Flow scheduler, we investigated the estimated noise $x_k^{(t,l)}$ generated by each transformer block at every step across 10,000 samples. Here, t represents the step, l denotes the block index, and k indicates the sample index. These samples included 5,000 prompts each from the MS-COCO 2014 validation split and MS-COCO 2017 validation split.

Evolution of Denoising Process. Based on multi-wavelet functions, the discrete wavelet transform (DWT) decomposes an input into four wavelet coefficients: LL, LH, HL, and HH. The LL coefficient captures the low-frequency component, reflecting structural features, while LH, HL, and HH coefficients detect high-frequency components associated with detail features. Initially, we apply DWT to each $x_k^{(t,l)}$ for decomposition into $x_k^{(t,l)} = \{(x_{k,LL}^{(t,l)}, x_{k,LH}^{(t,l)}, x_{k,HL}^{(t,l)}, x_{k,HH}^{(t,l)})\}$. Subsequently, we compute the L2 normalization for each subband of $x_k^{(t,l)}$. Finally, we calculate the maximum, minimum, and average values across batch and layer dimensions to derive $\|x^{(t)}\|_2$, which are used to generate the Figs. 2 (a) and (b).

Evolution of Feature Redundancies. Initially, we compute the cosine similarity for each token pair in $x_k^{(t,l)}$. Using the output similarity matrix, we identify the most similar token $x_{k,(i',j')}^{(t,l)}$ for every token $x_{k,(i,j)}^{(t,l)}$, where (i,j) and (i',j') are coordinates of a token and its most similar counterpart. We assess the location and degree of token redundancies by calculating the mean L2 distance and the cosine similarity among the closest pairs. In Fig. 2 (c), we calculate the L2 distance for each token pair (i,j) and (i',j'), averaging these distances across pairs and batches to compute the mean L2 distance for each layer and step. We then average these results across the layer dimension to track the evolution across timesteps. Similarly, in Fig. 2 (d), we compute the cosine similarity for each token pair, averaging these values across pairs and batches to derive the mean similarity for each layer and step and averaging across the layer dimension to track the evolution along timesteps.

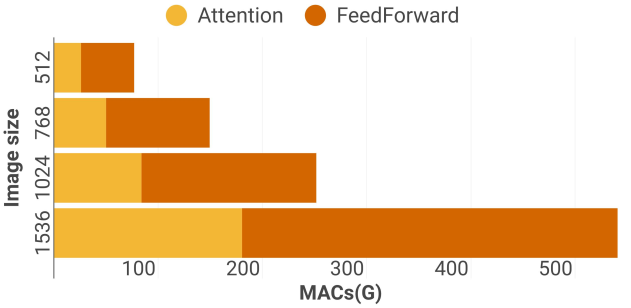

Figure S1. MACs of main components in different image sizes.

A.2. Details of Backbones, Ours, and SoTAs

We implement our methods and other SoTA techniques, including ToMeSD, AT-EDM, TokenCache, and DyDiT, on the SD3 Medium model for evaluation.

The SD3 Medium Backbone. The SD3 Medium employs an advanced transformer architecture, the multimodal diffusion transformer (MMDiT). Distinct from diffusion models that rely on U-Net architectures, it incorporates 24 JointTransformerBlocks at uniform feature levels, enabling joint attention interactions between prompt and image tokens. The SD3 model employs the Rectified Flow scheduler with a default CFG of 7.0 and 50 / 28 denoising steps. Fig. S1 illustrates the MACs of each computational component within the JointTransformerBlocks. It is observed that the FeedForward accounts for approximately $2/3$ of the MACs, while the Attention mechanism occupies about $1/3$. In the following, we implemented the SoTAs token reduction methods for DiT to reduce the computation.

Ours. We implement our method in the SD3 Medium using a two-stage approach. In the early stage, we execute similarity-prioritized structure merging before the MHSA and MLP blocks. In the later stage, we perform inattentive-prioritized detail merging prior to the MLP blocks. Additionally, we dynamically adjust the compression ratio and prompt weights at each denoising step using compression ratio adjusting and prompt token reweighting. Depending on the settings of CRA, we differentia between dynamic ratio and adaptive threshold, resulting in two variants: `SDTM` and `SDTM`*. During the batch inferences of `SDTM`*, if the batch size exceeds one, inconsistencies in token numbers across the batch may occur due to adaptive threshold filtering. To mitigate this, we balance the difficulty of prompts within each batch using GPT scores and select the minimum number of tokens for pruning.

ToMeSD. ToMeSD employs a uniform merging strategy across all sampling steps, utilizing a 2D stride-based strategy to merge tokens. This strategy is applied on blocks at the highest-resolution feature level. Since all blocks in the MMDiT architecture share the same feature level, we initially experimented with applying ToMeSD across all Transformer blocks. However, this led to significant degradation in generation quality. Given that ToMeSD was originally designed for UNet-based DMs, where convolution and transformer modules alternate (and thus not all modules incorporate token merging), indiscriminate application of token merging to all Transformer modules in MMDiT proved suboptimal. To address this, we adapted the application of ToMeSD for MMDiT by implementing a staggered compression pattern: for every four consecutive Transformer blocks, we apply no compression, standard compression, reduced compression, and standard compression rates, respectively. This configuration achieve better generation quality in equivalent overall cost. Furthermore, we implemented ToMeSD's linearly interpolating of the ratio.

AT-EDM. AT-EDM adopts a two-stage token pruning strategy, where $T - 0.7T$ serves as the early stage and $0.7T - 0$ as the later stage. This method uses a graph-based algorithm for token pruning across multiple cascaded attention block groups. We structured the MMDiT into six groups to align with this configuration. Additionally, AT-EDM incorporates a DSAP schedule that preserves unpruned attention blocks at the mid-stage due to the low feature level; this concern is not present in MMDiT. AT-EDM leaves the first attention block in each down-stage and the last in each up-stage unpruned. Consequently, we designate the first three groups as down-stage and the last three as up-stage to mirror this architecture.

TokenCache. TokenCache deploys a TPRR token pruning schedule, where $T - M$ serves as Phase I and $M - 0$ as Phase II. During each phase, a cyclic schedule is employed, alternating between I-steps, with no pruning, and K-steps, where a Cache Predictor is trained to prune non-essential tokens. We implement TokenCache in SD3 Medium using this strategy and train the Cache Predictor according to the procedures outlined in their paper. We determine the optimal values of (M, K_1, K_2), based on their ablation study, to be $(0.5T, 4, 2)$, where K_1 and K_2 correspond to different steps for Phase I and Phase II.

DyDiT. DyDiT introduces a timestep-wise dynamic width (TDW) to reduce model width and a spatialwise dynamic token (SDT) strategy to minimize redundancy at spatial locations. We integrated these strategies into SD3 Medium, using the FLOPs-aware end-to-end training method proposed by them to train the model. To ensure training stability, DyDiT incorporates fine-tuning stabilization, which is crucial for the FLOPs-aware end-to-end training method. Consequently, we conducted four training sessions and selected the optimal results. For the hyperparameter of cache interval, we chose an interval of 2, as identified as optimal in their ablation study.

A.3. Details of Evaluations

Our evaluation adheres to the settings employed in AT-EDM, conducting experiments on the COCO2017 validation split. We implemented a prompt deduplication strategy to ensure unique pairings of each image in the validation set with one prompt. Each image is center-cropped and resized for comparison. For metric calculations, we utilize the clean-fid[1] to compute FID scores and the ViT-G/14 model from Open-CLIP[2] to calculate CLIP scores. Unless specified otherwise, all experiments were conducted using two NVIDIA A100 GPUs, generating images of 1024×1024 resolution with a batch size of four.

B. Additional Analyses

B.1. Reduced Computation via Timesteps

In our methodology, we introduce similarity-prioritized structure merging to enhance the efficiency of MHSA and MLP blocks during the structure stage, and inattentive-prioritized detail merging to speed up MLP blocks in the detail stage. Additionally, we progressively decrease the compression ratio as denoising progresses. Based on the SD3 Medium model with our `SDTM` approach, we generate images with a resolution of 1024×1024 and display the MACs of MHSA and MLP blocks at representative steps 40, 30, 20, and 10 in Fig. S2. The results indicate increasing computational allocation across the timesteps, aligning with the evolution of feature redundancies over time as discussed in Sec. 3.

[1] https://github.com/GaParmar/clean-fid/tree/main
[2] https://github.com/mlfoundations/open-clip

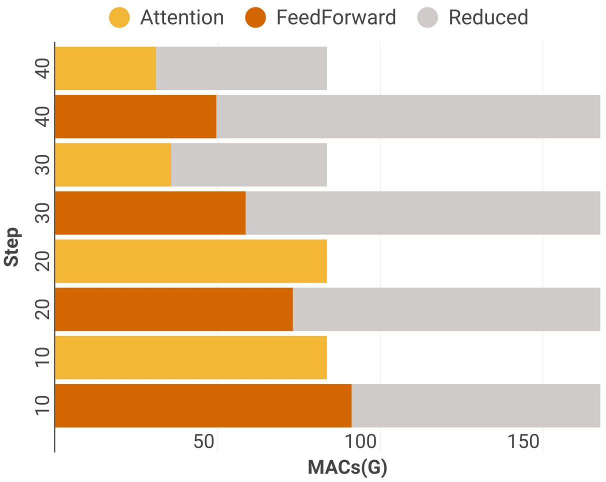

Figure S2. Reduced MACs of main components along timesteps.

Object	Complexity	SD3-M MACs	SD3-L MACs
Model	-	6.01T	20.13T
Similarity	$\mathcal{O}(ND + \frac{N}{m^2}\log(\frac{N}{m^2}))$	3.0E-04T	1.0E-03T
Inattentive	$\mathcal{O}(N^2 + N\log N)$	8.1E-04T	2.7E-03T

Table S1. Complexity and cost analysis of identifying redundancy.

B.2. Identification Computation

Although similarity-prioritized structure merging and inattentive-prioritized detail merging reduce the computation of original transformer blocks, they introduce additional cost for identifying similarity and inattentive redundancy. Assuming the feature $X \in \mathbb{R}^{N,D}$ and a window size $m \times m$, we report the complexity and MACs for redundancy identification in Table S1. Results show that for both the SD3 Medium and SD3.5 Large models, the computational cost of redundancy identification is negligible compared to the overall model cost.

C. Additional Results

C.1. More Comparisons with Baselines

More Comparisons with Baselines. We expanded the integration of SDTM and SDTM* into additional baselines to assess their adaptability. These included the SD3.5 Large Turbo, a distilled version of SD3.5 Large designed to enhance image quality with fewer denoising steps[3]; and FLUX.1-dev, a 12 billion-parameter rectified flow transformer noted for its advanced performance in image generation. We utilized their default CFG values and recommended schedulers. For FLUX.1-dev, we observed that including Rotary Positional Embedding in the MHSA blocks is extremely sensitive to token reduction; therefore, we left these MHSA blocks unpruned. The outcomes, presented in Table S2, indicate that despite considerable reductions in steps and the constraints imposed on MHSA, which lower the compression ratio, our method still achieves a favorable acceleration while maintaining the image quality.

Method	Step	W-MACs(T) ↓	Latency(s) ↓	FID ↓
SD3.5 Large Turbo	4	47.7	0.69	30.48
+SDTM	4	33.4	0.53	31.25
+SDTM*	4	32.8	0.52	30.62
SD3.5 Large Turbo	10	119.4	1.30	30.27
+SDTM	10	75.6	0.89	31.01
+SDTM*	10	74.5	0.87	30.41
FLUX.1-dev	20	595.2	13.81	30.25
+SDTM†	20	386.5	10.15	30.68
+SDTM*†	20	381.4	10.08	30.21

Table S2. Comparisons of our SDTM and SDTM with SD3 Large Turbo and FLUX.1-dev at various steps. A dagger † indicates that the MHSA block is not accelerated for adaptation.

Method	Image size	MACs(T) ↓	Latency(s) ↓	FID ↓
baseline	512	1.83	2.47	28.74
+SDTM	512	1.10	1.61	29.86
baseline	768	3.58	5.39	29.27
+SDTM	768	2.16	3.54	29.65
baseline	1536	12.97	31.29	28.06
+SDTM	1536	8.04	21.28	28.24

Table S3. Ablation of our SDTM with the SD3 Medium using a 50-RF scheduler across various image sizes.

C.2. Ablation of Image Size

Following the settings of AT-EDM, our experiments were primarily conducted on 1024 px images. To assess our method's applicability across different image sizes, we tested resolutions including 512, 768, and 1536 px, with results detailed in Table S3. We utilized the SD3 Medium equipped with a 50-RF scheduler as our baseline. It is important to note that the FID scores for different image sizes are not comparable since they correspond to distinct distributions. The results demonstrate that smaller image sizes slightly compromise image quality; this reduction can be attributed to the decreased similarity between patches at smaller sizes, which leads to less feature redundancy. Nonetheless, our approach consistently delivers significant acceleration compared to the baseline method.

C.3. More Visualization of Samples

In Fig. S3, we compare our methods with fine-tuning-free techniques including ToMeSD and AT-EDM based on the SD3.5 Large. Fig. S4 displays uncurated images generated by our methods across SD3 Medium and SD3.5 Large. Our methods outperform other SoTA techniques and maintain robust generative capabilities across various scenes.

[3] https://stability.ai/news/introducing-stable-diffusion-3-5

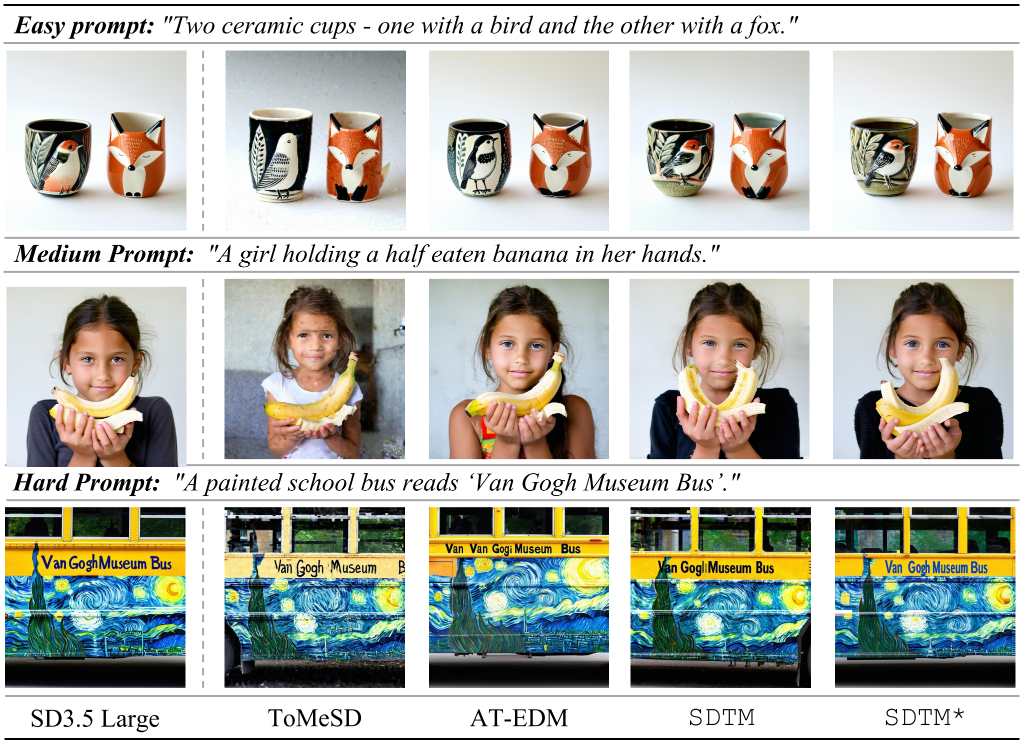
Easy prompt: "Two ceramic cups - one with a bird and the other with a fox."
Medium Prompt: "A girl holding a half eaten banana in her hands."
Hard Prompt: "A painted school bus reads 'Van Gogh Museum Bus'."
SD3.5 Large
ToMeSD
AT-EDM
SDTM
SDTM*

Figure S3. Qualitative comparison on SD3.5 Large under varying data complexities. For ToMeSD and AT-EDM, we use versions with approximately 1.3× acceleration, while others use approximately 1.5× versions. Best viewed when zoomed in.

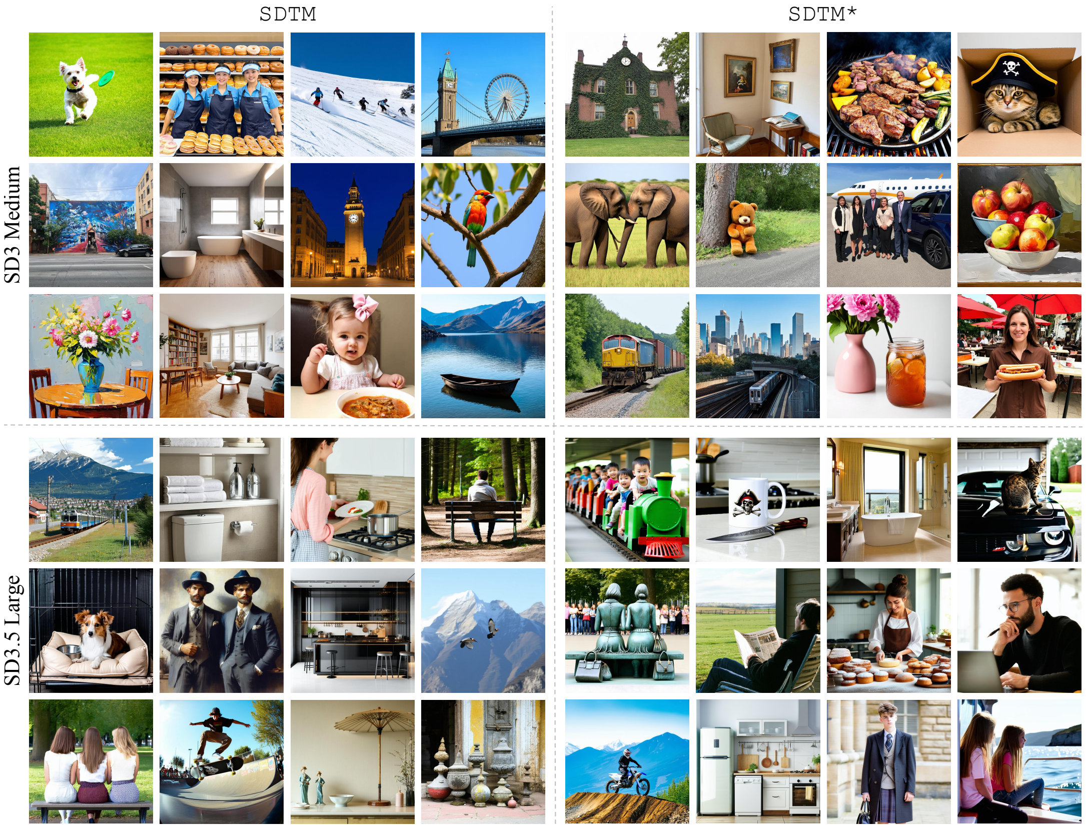

Figure S4. Uncurated images generated using SD3 Medium and SD3.5 Large configurations under the `SDTM` and `SDTM*` frameworks.