Attention-Driven Training-Free Efficiency Enhancement of Diffusion Models

Hongjie Wang[1]*, Difan Liu[2], Yan Kang[2], Yijun Li[2], Zhe Lin[2], Niraj K. Jha[1], Yuchen Liu[2]†
[1]Princeton University, [2]Adobe Research

Abstract

*Diffusion Models (DMs) have exhibited superior performance in generating high-quality and diverse images. However, this exceptional performance comes at the cost of expensive architectural design, particularly due to the attention module heavily used in leading models. Existing works mainly adopt a retraining process to enhance DM efficiency. This is computationally expensive and not very scalable. To this end, we introduce the **A**ttention-driven **T**raining-free **E**fficient **D**iffusion **M**odel (**AT-EDM**) framework that leverages attention maps to perform run-time pruning of redundant tokens, without the need for any retraining. Specifically, for single-denoising-step pruning, we develop a novel ranking algorithm, Generalized Weighted Page Rank (G-WPR), to identify redundant tokens, and a similarity-based recovery method to restore tokens for the convolution operation. In addition, we propose a Denoising-Steps-Aware Pruning (DSAP) approach to adjust the pruning budget across different denoising timesteps for better generation quality. Extensive evaluations show that AT-EDM performs favorably against prior art in terms of efficiency (e.g., 38.8% FLOPs saving and up to 1.53× speed-up over Stable Diffusion XL) while maintaining nearly the same FID and CLIP scores as the full model. Project webpage: https://atedm.github.io.*

1. Introduction

Diffusion Models (DMs) [9, 29] have revolutionized computer vision research by achieving state-of-the-art performance in various text-guided content generation tasks, including image generation [28], image editing [12], super resolution [17], 3D objects generation [27], and video generation [10]. Nonetheless, the superior performance of DMs comes at the cost of an enormous computation budget. Although Latent Diffusion Models (LDMs) [28, 34] make text-to-image generation much more practical and affordable for normal users, their inference process is still too slow. For example, on the current flagship mobile phone, generating a single 512px image requires 90 seconds [19].

To address this issue, numerous approaches geared at efficient DMs have been introduced, which can be roughly categorized into two regimes: (1) efficient sampling strategy [24, 30] and (2) efficient model architecture [19, 38]. While efficient sampling methods can reduce the number of denoising steps, they do not reduce the memory footprint and compute cost for each step, making it still challenging to use on devices with limited computational resources. On the contrary, an efficient architecture reduces the cost of each step and can be further combined with sampling strategies to achieve even better efficiency. However, most prior efficient architecture works **require retraining** of the DM backbone, which can take thousands of A100 GPU hours. Moreover, due to different deployment settings on various platforms, different compression ratios of the backbone model are required, which necessitate multiple retraining runs later. Such retraining costs are a big concern even for large companies in the industry.

To this end, we propose the **Attention-driven Training-free Efficient Diffusion Model (AT-EDM)** framework, which accelerates DM inference at run-time without any retraining. To the best of our knowledge, training-free architectural compression of DMs is a highly uncharted area. Only one prior work, Token Merging (ToMe) [1], addresses this problem. While ToMe demonstrates good performance on Vision Transformer (ViT) acceleration [2], its performance on DMs still has room to improve. To further enrich research on training-free DM acceleration, we start our study by profiling the floating-point operations (FLOPs) of the state-of-the-art model, Stable Diffusion XL (SD-XL) [26], through which we find that attention blocks are the dominant workload. In a single denoising step, we thus propose to dynamically prune redundant tokens to accelerate attention blocks. We pioneer a fast graph-based algorithm, Generalized Weighted Page Rank (G-WPR), inspired by Zero-TPrune [35], and deploy it on attention maps in DMs to identify superfluous tokens. Since SD-XL contains ResNet blocks, which require a full number of tokens for the convolution operations, we propose a novel similarity-based token copy approach to recover pruned tokens, again leveraging the rich information provided by the

*Work was partly done during an internship at Adobe.
†Corresponding Author.

Figure 1. Examples of applying AT-EDM to SD-XL [26]. Compared to the full-size model (**top row**), our accelerated model (**bottom row**) has around 40% FLOPs reduction while enjoying competitive generation quality at various aspect ratios.

attention maps. This token recovery method is critical to maintaining image quality. We find that naive interpolation or padding of pruned tokens adversely impacts generation quality severely. In addition to single-step token pruning, we also investigate cross-step redundancy in the denoising process by analyzing the variance of attention maps. This leads us to a novel pruning schedule, dubbed as Denoising-Steps-Aware Pruning (DSAP), in which we adjust the pruning ratios across different denoising timesteps. We find DSAP not only significantly improves our method, but also helps improve other run-time pruning methods like ToMe [1]. Compared to ToMe, our approach shows a clear improvement by generating clearer objects with sharper details and better text-image alignment under the same acceleration ratio. In summary, our contributions are four-fold:

- We propose the AT-EDM framework, which leverages rich information from attention maps to accelerate pre-trained DMs without retraining.
- We design a token pruning algorithm for a single denoising step. We pioneer a fast graph-based algorithm, G-WPR, to identify redundant tokens, and a novel similarity-based copy method to recover missing tokens for convolution.
- Inspired by the variance trend of attention maps across denoising steps, we develop the DSAP schedule, which improves generation quality by a clear margin. The schedule also provides improvements over other run-time acceleration approaches, demonstrating its wide applicability.
- We use AT-EDM to accelerate a top-tier DM, SD-XL, and conduct both qualitative and quantitative evaluations. Noticeably, our method shows comparable performance with an FID score of 28.0 with 40% FLOPs reduction relative to the full-size SD-XL (FID 27.3), achieving state-of-the-art results. Visual examples are shown in Fig. 1.

2. Related Work

Text-to-Image Diffusion Models. DMs learn to reverse the diffusion process by denoising samples from a normal distribution step by step. In this manner, the diffusion-based generative models enable high-fidelity image synthesis with variant text prompts [4, 9]. However, DMs in the pixel space suffer from large generation latency, which severely limits their applications [36]. The LDM [28] was the first to train a Variational Auto-Encoder (VAE) to encode the pixel space into a latent space and apply the DM to the latent space. This reduces computational cost significantly while maintaining generation quality, thus greatly enhancing the application of DMs. Subsequently, several improved versions of the LDM, called Stable Diffusion Models (SDMs), have been released. The most recent and powerful open-source version is SD-XL [26], which outperforms previous versions by a large margin. SD-XL is our default backbone in this work.

Efficient Diffusion Models. Researchers have made enormous efforts to make DMs more efficient. Existing efficient DMs can be divided into two types:

(1) **Efficient sampling** to reduce the required number of denoising steps [22, 30–32]. A recent efficient sampling work [24] managed to reduce the number of denoising steps to as low as one. It achieves this by iterative distillation, halving the number of denoising steps each time.

(2) **Architectural compression** to make each sampling step more efficient [11, 19, 36, 38]. A recent work [13] removes multiple ResNet and attention blocks in the U-Net through distillation. Although these methods can save computational costs while maintaining decent image quality, they require *retraining* of the DM backbone to enhance efficiency, needing thousands of A100 GPU hours. Thus, a training-free method to enhance the efficiency of DMs is needed. Note that our proposed training-free framework, AT-EDM,

is **orthogonal** to these efficiency enhancement methods and can be stacked with them to further improve their efficiency. We provide corresponding experimental evidence in Supplementary Material.

Training-Free Efficiency Enhancement. Training-free (i.e., post-training) efficiency enhancement schemes have been widely explored for CNNs [14, 33, 39] and ViTs [2, 7, 15, 35]. However, training-free schemes for DMs are still poorly explored. To the best of our knowledge, the only prior work in this field is ToMe [1]. It uses token embedding vectors to obtain pair-wise similarity and merges similar tokens to reduce computational overheads. While ToMe achieves a decent speed-up when applied to SD-v1.x and SD-v2.x, we find that it does not help much when applied to the state-of-the-art DM backbone, SD-XL, whilst our method achieves a clear improvement over it (see experimental results in Section 4). This is mainly due to (1) the significant architectural change of SD-XL (see Supplementary Material); (2) our better algorithm design to identify redundant tokens.

Exploiting Attention Maps. We aim to design a method that exploits information present in pre-trained models. ToMe only uses embedding vectors of tokens and ignores the correlation between tokens. We take inspiration from recent image editing works [3, 5, 8, 25], in which attention maps clearly demonstrate which parts of a generated image are more important. This inspires us to use the correlations and couplings between tokens indicated by attention maps to identify unimportant tokens and prune them. Specifically, we can convert attention maps to directed graphs, where nodes represent tokens, without information loss. Based on this idea, we develop the G-WPR algorithm for token pruning in a single denoising step.

Non-Uniform Denoising Steps. Various existing works [6, 18, 21, 37] demonstrate that denoising steps contribute differently to the quality of generated images; thus, it is not optimum to use uniform denoising steps. OMS-DPM [21] builds a model zoo and uses different models in different denoising steps. It trains a performance predictor to assist in searching for the optimal model schedule. DDSM [37] employs a spectrum of neural networks and adapts their sizes to the importance of each denoising step. AutoDiffusion [18] employs evolutionary search to skip some denoising steps and some blocks in the U-Net. Diff-Pruning [6] uses a Taylor expansion over pruned timesteps to disregard non-contributory diffusion steps. All existing methods either require an intensive training/fine-tuning/searching process to obtain and deploy the desired denoising schedule or are not compatible with our proposed G-WPR token pruning algorithm due to the U-Net architecture change. On the contrary, based on our investigation of the variance of attention maps across denoising steps, we propose DSAP. Its schedule can be determined via simple ablation experiments and it is compatible with any token pruning scheme. DSAP can potentially be migrated to existing efficient DMs to help improve their image quality.

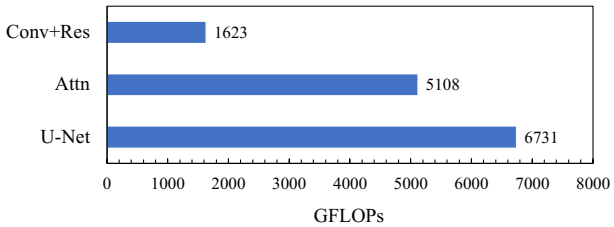


Figure 2. U-Net FLOPs breakdown of SD-XL [26] measured with 1024px image generation. Among components of U-Net (convolution blocks, ResNet blocks, and attention blocks), attention blocks cost the most.

3. Methodology

We start our investigation by profiling the FLOPs of the state-of-the-art DM, SD-XL, as shown in Fig. 2. Noticeably, among compositions of the sampling module (U-Net), attention blocks, which consist of several consecutive attention layers, dominate the workload for image generation. Therefore, we propose AT-EDM to accelerate attention blocks in the model through token pruning. AT-EDM contains two important parts: a single-denoising-step token pruning scheme and the DSAP schedule. We provide an overview of these two parts and then discuss them in detail.

3.1. Overview

Fig. 3 illustrates the two main parts of AT-EDM:

Part I: Token pruning scheme in a single denoising step.
Step 1: We obtain the attention maps from an attention layer in the U-Net. We can potentially obtain the attention maps from self-attention or cross-attention. We compare the two choices and analyze them in detail through ablation experiments.
Step 2: We use a scoring module to assign an importance score to each token based on the obtained attention map. We use an algorithm called G-WPR to assign importance scores to each token. This is described in Section 3.2.
Step 3: We generate pruning masks based on the calculated importance score distribution. Currently, we simply use the top-k approach to determine the retained tokens, i.e., prune tokens with less importance scores.
Step 4: We use the generated mask to perform token pruning. We do this after the feed-forward layer of attention layers. We may also perform pruning early before the feed-forward layers. We provide ablative experimental results for it in Supplementary Material.
Step 5: We repeat Steps 1-4 for each consecutive attention layer. Note that we do not apply pruning to the last attention layer before the ResNet layer.

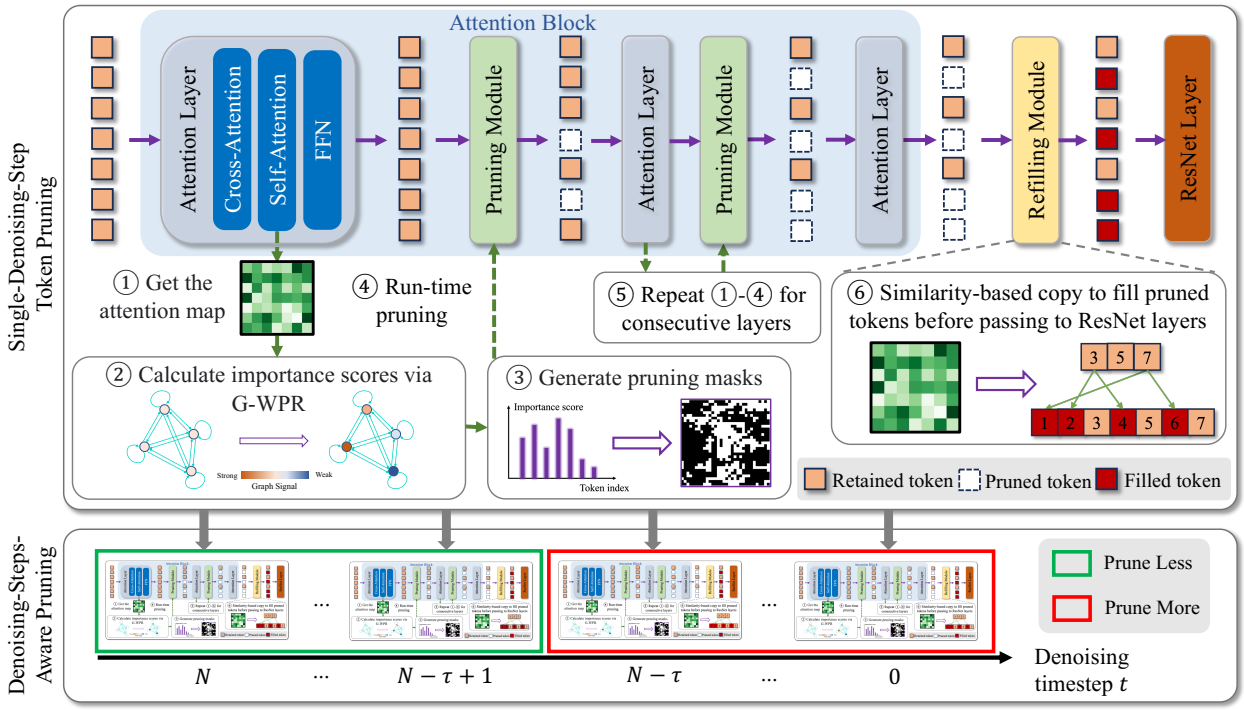


Figure 3. Overview of our proposed efficiency enhancement framework **AT-EDM**. **Single-Denoising-Step Token Pruning:** (1) We get the attention map from self-attention. (2) We calculate the importance score for each token using G-WPR. (3) We generate pruning masks. (4) We apply the masks to tokens after the feed-forward network to realize token pruning. (5) We repeat Steps (1)-(4) for each consecutive attention layer. (6) Before passing feature maps to the ResNet block, we recover pruned tokens through similarity-based copy. **Denoising-Steps-Aware Pruning Schedule:** In early steps, we propose to prune fewer tokens and to have less FLOPs reduction. In later steps, we prune more aggressively for higher speedup.

Step 6: Finally, before passing the pruned feature map to the ResNet block, we need to fill (i.e., try to recover) the pruned tokens. A simple approach is to pad zeros, which means we do not fill anything. The method that we currently use is to copy tokens to corresponding locations based on similarity. This is described in detail in Section 3.2.

Part II: DSAP schedule. Attention maps in early denoising steps are more chaotic and less informative than those in later steps, which is indicated by their low variance. Thus, they have a weaker ability to differentiate unimportant tokens [8]. Based on this intuition, we design the DSAP schedule that prunes fewer tokens in early denoising steps. Specifically, we select some attention blocks in the up-sampling and down-sampling stages and leave them unpruned, since they contribute more to the generated image quality than other attention blocks [19]. We demonstrate the schedule in detail in Section 3.3.

3.2. Part I: Token Pruning in a Single Step

Notation. Suppose $\mathbf{A}^{(h,l)} \in \mathbb{R}^{M\times N}$ is the attention map of the h-th head in the l-th layer. It reflects the correlations between M Query tokens and N Key tokens. We refer to $\mathbf{A}^{(h,l)}$ as $\mathbf{A}$ for simplicity in the following discussion. Let $A_{i,j}$ denote its element in the i-th row, j-th column. $\mathbf{A}$ can be thought of as the adjacency matrix of a directed graph in the G-WPR algorithm. In this graph, the set of nodes with input (output) edges is referred to as Φ_{in} (Φ_{out}). Nodes in Φ_{in} (Φ_{out}) represent Key (Query) tokens, i.e., $\Phi_{in} = \{k_j\}_{j=1}^{N}$ ($\Phi_{out} = \{q_i\}_{i=1}^{M}$). Let s_K^t (s_Q^t) denote the vector that represents the importance score of Key (Query) tokens in the t-th iteration of the G-WPR algorithm. In the case of self-attention, Query tokens are the same as Key tokens. Specifically, we let $\{x_i\}_{i=1}^{N}$ denote the N tokens and s denote their importance scores in the description of our token recovery method.

The G-WPR Algorithm. WPR [35] uses the attention map as an adjacency matrix of a directed complete graph. It uses a graph signal to represent the importance score distribution among nodes in this graph. This signal is initialized uniformly. WPR uses the adjacency matrix as a graph operator, applying it to the graph signal iteratively until convergence. In each iteration, each node votes for which node is more important. The weight of the vote is determined by its importance in the last iteration. However, WPR, as proposed in [35], constrains the used attention map to be a self-attention map. Based on this, we propose the G-WPR algorithm, which is compatible with both self-attention and cross-attention, as shown in Algorithm 1. The attention

from Query q_i to Key k_j weights the edge from q_i to k_j in the graph generated by $\mathbf{A}$. In each iteration of the vanilla WPR, by multiplying with the attention map, we map the importance of Query tokens s_Q^t to the importance of Key tokens s_K^{t+1}, i.e., each node in Φ_{out} votes for which Φ_{in} node is more important. For self-attention, $s_Q^{t+1} = s_K^{t+1}$ since Query and Key tokens are the same. For cross-attention, Query tokens are image tokens and Key tokens are text prompt tokens. Based on the intuition that important image tokens should devote a large portion of their attention to important text prompt tokens, we define function $f(\mathbf{A}, s_K)$ that maps s_K^{t+1} to s_Q^{t+1}. One entropy-based implementation is

$$s_Q^{t+1}(q_i) = f(\mathbf{A}, s_K^{t+1}) = \frac{\sum_{j=1}^{N} A_{i,j} \cdot s_K^{t+1}(k_j)}{-\sum_{j=1}^{N} A_{i,j} \cdot \ln A_{i,j}} \quad (1)$$

where $A_{i,j}$ is the attention from Query q_i to Key k_j. This is the default setting for cross-attention-based WPR in the following sections. We discuss and compare other implementations in Supplementary Material. Note that for self-attention, $f(\mathbf{A}, s_K^{t+1}) = s_K^{t+1}$. The G-WPR algorithm has an $O(M \times N)$ complexity, where M (N) is the number of Query (Key) tokens. We employ this algorithm in each head and then obtain the root mean square of scores from different heads (to reward tokens that obtain very high importance scores in a few heads).

Algorithm 1 The G-WPR algorithm for both self-attention and cross-attention

Require: $M, N > 0$ is the number of nodes in Φ_{out}, Φ_{in}; $\mathbf{A} \in \mathbb{R}^{M \times N}$; $s_Q \in \mathbb{R}^M, s_K \in \mathbb{R}^N$; $f(\mathbf{A}, s_k)$ maps the importance of Key to that of Query

Ensure: $s \in \mathbb{R}^M$ represents the importance score of image tokens

$s_Q^0 \leftarrow \frac{1}{M} \times e_M$
$t \leftarrow 0$
while $(|s_Q^t - s_Q^{t-1}| > \epsilon)$ **or** $(t = 0)$ **do**
  $s_K^{t+1} \leftarrow \mathbf{A}^T \times s_Q^t$
  $s_Q^{t+1} \leftarrow f(\mathbf{A}, s_K^{t+1})$
  $s_Q^{t+1} \leftarrow s_Q^{t+1} / |s_Q^{t+1}|$
  $t \leftarrow t + 1$
end while
$s \leftarrow s_Q^t$

Recovering Pruned Tokens. We have fewer tokens after token pruning, leading to efficiency enhancement. However, retained tokens form irregular maps and thus cannot be used for convolution, as shown in Fig. 4. We need to recover the pruned tokens to make them compatible with the following convolutional operations in the ResNet layer.

(I) Padding Zeros. One straightforward way to do this is to pad zeros. However, to maintain the high quality of generated images, we hope to recover the pruned tokens as precisely as possible, as if they were not pruned.

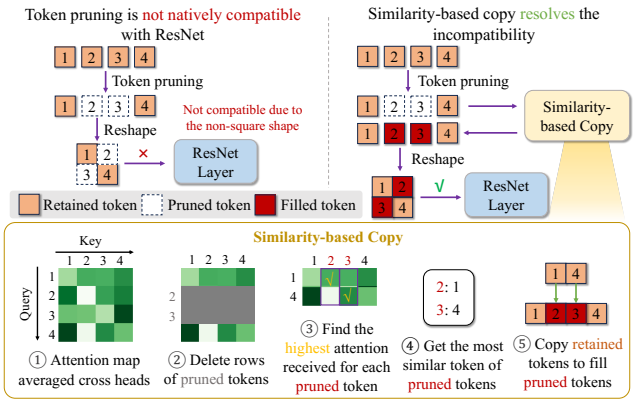


Figure 4. Our similarity-based copy method for token recovering resolves the incompatibility between token pruning and ResNet. Token pruning incurs the non-square shape of feature maps and thus is not compatible with ResNet. To address this issue, we propose similarity-based copy to recover the pruned tokens. It first averages the attention map across heads and deletes the rows of pruned tokens to avoid selecting them as the most similar one. Then, it finds the source of the highest attention received for each pruned token and copies the corresponding retained tokens for recovery. After recovering, the tokens can be translated into a spatially-complete feature map to serve as input to ResNet blocks.

(II) Interpolation. Interpolation methods, such as bicubic interpolation, are not suitable in this context. To use the interpolation algorithm, we first pad zeros to fill the pruned tokens and form a feature map of size $N \times N$. Then we downsample it to $\frac{N}{2} \times \frac{N}{2}$ and upsample it back to $N \times N$ with the interpolation algorithm. We keep the values of retained tokens fixed and only use the interpolated values of pruned tokens. Due to the high pruning rates (usually larger than 50%), most tokens that represent the background get pruned, leading to lots of pruned tokens that are surrounded by other pruned tokens instead of retained tokens. Interpolation algorithms assign nearly zero values to these tokens.

(III) Direct copy. Another possible method is to use the corresponding values before pruning is applied (i.e., before being processed by the following attention layers) to fill the pruned tokens. The problem with this method is that the value distribution changes significantly after being processed by multiple attention layers, and copied values are far from the values of these tokens if they are not pruned and are processed by the following attention layers.

To avoid the effect of distribution shift, we propose the **similarity-based copy** technique, as shown in Fig. 4. Instead of copying values that are not processed by attention layers, we select tokens that are similar to pruned tokens from the retained tokens. We use the self-attention map to determine the source of the highest attention received for each pruned token and use that as the most similar one. This is based on the intuition that attention from token x_a to token x_b, $A_{a,b}$, is determined by two factors: (1) importance

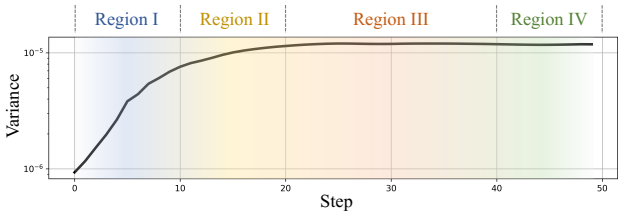


Figure 5. Variance of attention maps in different denoising steps. We divide the denoising steps into four typical regions: (I) Very-early steps: Variance of attention maps is small and increases rapidly. (II) Mid-early steps: Variance of attention maps is large and increases slowly. (III) Middle steps: Variance of attention maps is large and almost constant. (IV) Last several steps.

of token x_b, i.e., $s(x_b)$, and (2) similarity between token x_a and x_b. If we observe the attention that x_b receives, i.e., compare $\{A_{i,b}\}_{i\in N}$, since $s(x_b)$ is fixed, index $i = \eta$ that maximizes $\{A_{i,b}\}_{i\in N}$ is the index of the most similar token, i.e., x_η. Finally, we copy the value of token x_η to fill (i.e., recover) the pruned token x_b.

3.3. Part II: Denoising-Steps-Aware Pruning

Early denoising steps determine the layout of generated images and, thus, are crucial. On the contrary, late denoising steps aim at refining the generated image, natively including redundant computations since many regions of the image do not need refinement. In addition, *early denoising steps have a weaker ability to differentiate unimportant tokens*, and late denoising steps yield informative attention maps and differentiate unimportant tokens better. To support this claim, we investigate the variance of feature maps in different denoising steps, as shown in Fig. 5. It indicates that attention maps in early steps are more uniform. They assign similar attention scores to both important and unimportant tokens, making it harder to precisely identify unimportant tokens and prune them in early steps. Based on these intuitions, we propose DSAP that employs a **prune-less schedule** in early denoising steps by leaving some of the layers unpruned.

The Prune-Less Schedule. In SD-XL, each down-stage includes two attention blocks and each up-stage includes three attention blocks (except for stages without attention). The mid-stage also includes one attention block. Each attention block includes 2-10 attention layers. In our prune-less schedule, we select some attention blocks to not perform token pruning. Since previous works [13, 19] indicate that the mid-stage contributes much less to the generated image quality than the up-stages and down-stages, we do not select the attention block in the mid-stage. Based on the ablation study, we choose to leave the first attention block in each down-stage and the last attention block in each up-stage unpruned. We use this prune-less schedule for the first τ denoising steps. We explore setting τ in different regions shown in Fig. 5 and find $\tau = 15$ is the optimal choice. We present all the related ablative experimental results in Section 4.4. A detailed description of the less aggressive pruning schedule is provided in Supplementary Material. To further consolidate our intuitions, we also investigate a more aggressive pruning schedule in early denoising steps and find it is inferior to our current approach (see Supplementary Material).

4. Experimental Results

In this section, we evaluate AT-EDM and ToMe on SD-XL. We provide both visual and quantitative experimental results to demonstrate the advantages of AT-EDM over ToMe.

4.1. Experimental Setup

Common Settings. We implement both our AT-EDM method and ToMe on the official repository of SD-XL and evaluate their performance. The resolution of generated images is 1024×1024 pixels and the default FLOPs budget for each denoising step is assumed to be 4.1T, which is 38.8% smaller than that of the original model (6.7T) unless otherwise noted. The default CFG-scale for image generation is 7.0 unless otherwise noted. We set the total number of sampling steps to 50. We use the default sampler of SD-XL, i.e., EulerEDMSampler.

AT-EDM. For a concise design, we only insert a pruning layer after the first attention layer of each attention block and set the pruning ratio for that layer to ρ. To meet the FLOPs budget of 4.1T, we set $\rho = 63\%$. For the DSAP setting, we choose to leave the first attention block in each down-stage and the last attention block in each up-stage unpruned. We use this prune-less schedule for the first $\tau = 15$ denoising steps.

ToMe. The SD-XL architecture has changed significantly compared to previous versions of SDMs (see Supplementary Material). Thus, the default setting of ToMe does not lead to enough FLOPs savings. To meet the FLOPs budget, it is necessary to use a more aggressive merging setting. Therefore, we expand the application range of token merging (1) from attention layers at the highest feature level to all attention layers, and (2) from self-attention to self-attention, cross-attention, and the feedforward network. We set the merging ratio $r = 50\%$ to meet the FLOPs budget of 4.1T.

Evaluations. We first compare the generated images with manually designed challenging prompts in Section 4.2. Then, we report FID and CLIP scores of zero-shot image generation on the MS-COCO 2017 validation dataset [20] in Section 4.3. Tested models generate 1024×1024 px images based on the captions of 5k images in the validation set. We provide ablative experimental results and analyze them in Section 4.4 to justify our design choices. We provide more implementation details in Supplementary Material.

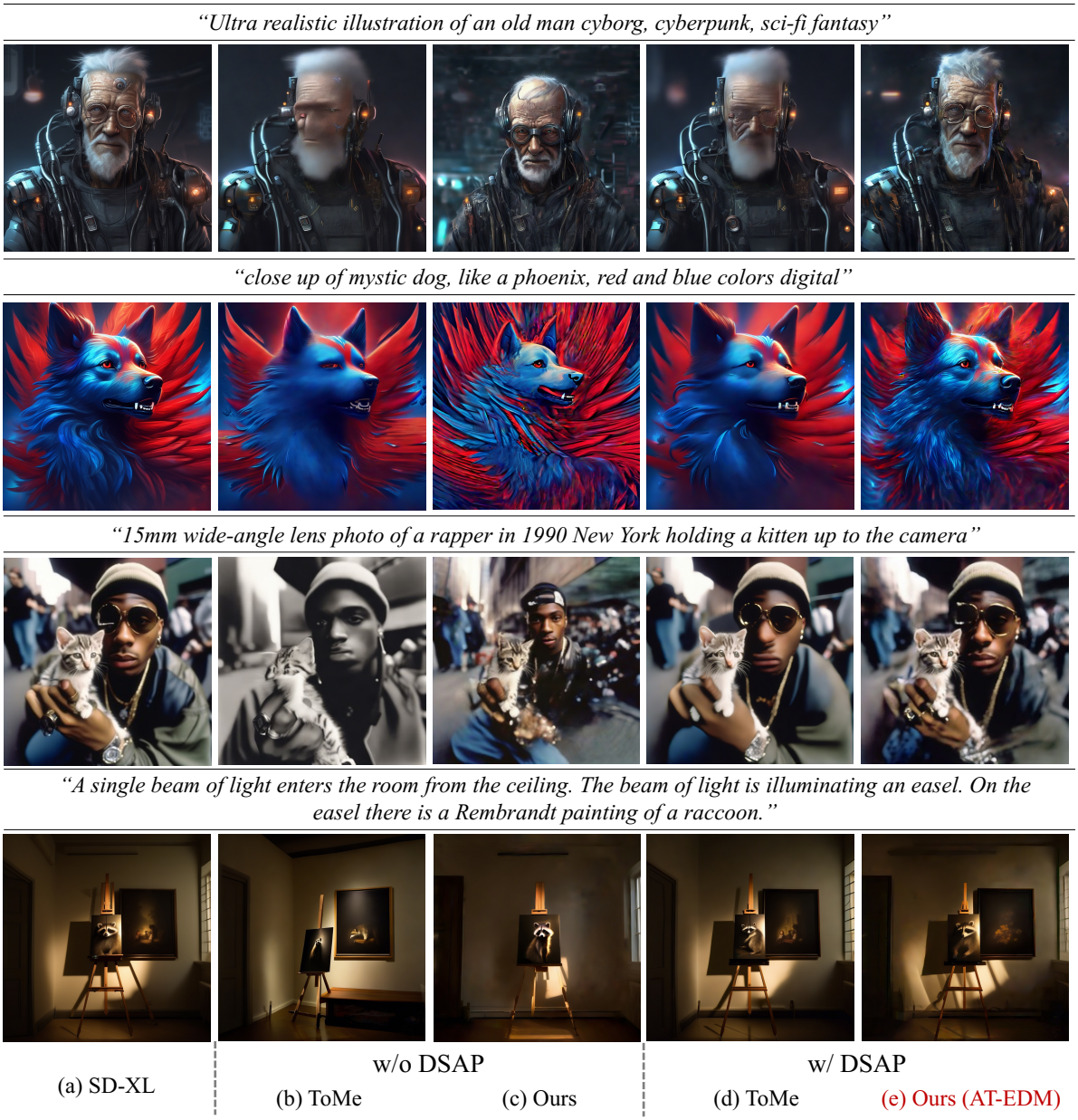


Figure 6. Comparing AT-EDM to the state-of-the-art approach, ToMe [2]. While the full-size SD-XL [26] (**Col. a**) consumes 6.7 TFLOPs, we compare the accelerated models (**Col. b-e**) at the same budget of 4.1 TFLOPs. Compared to ToMe, we find that AT-EDM's token pruning algorithm provides clearer generated objects with sharper details and finer textures, and a better text-image alignment where it better retains the semantics in the prompt (see the fourth row). Moreover, we find that DSAP provides better structural layout of the generated images, which is effective for both ToMe and our approach. AT-EDM combines the novel token pruning algorithm and the DSAP schedule (**Col. e**), outperforming the state of the art.

4.2. Visual Examples for Qualitative Analysis

We use manually designed challenging prompts to evaluate ToMe and our proposed AT-EDM framework. The generated images are compared in Fig. 6. We compare more generated images in Supplementary Material. Visual examples indicate that with the same FLOPs budget, AT-EDM demonstrates better **main object preservation** and **text-image alignment** than ToMe. For instance, in the first example, AT-EDM preserves the main object, the face of the old man, much better than ToMe does. AT-EDM's strong ability to preserve the main object is also exhibited in the second example. ToMe loses high-frequency features of the main object, such as texture and hair, while AT-EDM retains them well, even without DSAP. The third example again illustrates the advantage of AT-EDM over ToMe in preserving the rapper's face. The fourth example uses a relatively

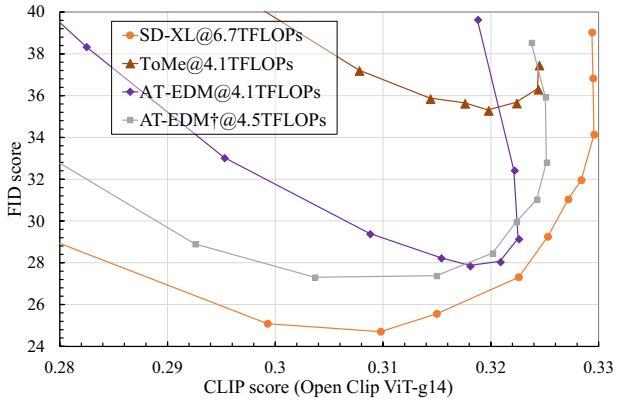


Figure 7. FID-CLIP score curves. The used CFG scales are [1.0, 1.5, 2.0, 2.5, 3.0, 4.0, 5.0, 6.0, 7.0, 9.0, 12.0, 15.0]. This figure is zoomed in to the bottom-right corner to show the comparison between the best trade-off points. AT-EDM outperforms ToMe by a clear margin. See complete curves in Supplementary Material.

complex prompt that describes relationships between multiple objects. ToMe misunderstands *"a Rembrandt painting of a raccoon"* as being a random painting on the easel and a painting of a raccoon on the wall. On the contrary, the image generated by AT-EDM understands and preserves these relationships very well, even without DSAP. As a part of our AT-EDM framework, DSAP is not only effective in AT-EDM but also beneficial to ToMe in improving image quality and text-image alignment. When we deploy DSAP in ToMe, we select corresponding attention blocks to not perform token merging, while keeping the FLOPs cost fixed.

4.3. Quantitative Evaluations

FID-CLIP Curves. We explore the trade-off between the CLIP and FID scores through various Classifer-Free Guidance (CFG) scales. We show the results in Fig. 7. AT-EDM† does not deploy pruning at the second feature level (see Supplementary Material). It indicates that for most CFG scales, AT-EDM not only lowers the FID score but also results in higher CLIP scores than ToMe, implying that images generated by AT-EDM not only have better quality but also better text-image alignment. Specifically, when the CFG scale equals 7.0, AT-EDM results in [FID, CLIP] = [28.0, 0.321], which is almost the same as the full-size one ([27.3, 0.323], `CFG_scale`=4.0). For comparison, ToMe results in [35.3, 0.320] with a CFG scale of 7.0. Thus, **AT-EDM reduces the FID gap from 8.0 to 0.7.**

Various FLOPs Budgets. We deploy ToMe and AT-EDM on SD-XL under various FLOPs budgets and quantitatively compare their performance in Table 1. The FLOPs cost in this table refers to the average FLOPs cost of a denoising step. Table 1 indicates that AT-EDM achieves better image quality than ToMe (lower FID scores) under all FLOPs budgets. When the FLOPs budget is extremely low (less than 50% of the full model), ToMe achieves higher CLIP scores than AT-EDM. When the FLOPs saving is 30-40%, AT-EDM achieves not only better image quality (lower FID scores) but also better text-image alignment (higher CLIP scores) than ToMe. Note that under the same CFG-scale, AT-EDM achieves a lower FID score than the full-size model while reducing FLOPs by 32.8%. In the case that it trades text-image alignment for image quality (via reducing the CFG scale to 4.0), AT-EDM achieves **not only a lower FID score but also a higher CLIP score than the full-size model while reducing FLOPs by 32.8%**. We provide more visual examples under various FLOPs budgets in Supplementary Material.

Table 1. Deploying ToMe and AT-EDM in SD-XL under different FLOPs budgets. We generate all images with the CFG-scale of 7.0, except for SD-XL†, for which we use a CFG-scale of 4.0.

Model	FID	CLIP	TFLOPs
SD-XL	31.94	0.3284	6.7
SD-XL†	27.30	0.3226	6.7
ToMe-a	58.76	0.2954	2.9
AT-EDM-a	52.00	0.2784	2.9
ToMe-b	40.94	0.3154	3.6
AT-EDM-b	29.80	0.3095	3.6
ToMe-c	35.27	0.3198	4.1
AT-EDM-c	28.04	0.3209	4.1
ToMe-d	32.46	0.3235	4.6
AT-EDM-d	27.23	0.3245	4.5

Latency Analysis. SD-XL uses the Fused Operation (FO) library, xformers [16], to boost its generation. The Current Implementation (CI) of xformers does not provide attention maps as intermediate results; hence, we need to additionally calculate the attention maps. We discuss the sampling latency for three cases: (I) without FO, (II) with FO under CI, and (III) with FO under the Desired Implementation (DI), which provides attention maps as intermediate results. Table 2 shows that with FO, the cost of deploying pruning at the second feature level exceeds the latency reduction it leads to. Hence, AT-EDM† is faster than AT-EDM. Fig. 8 shows the extra latency incurred by different pruning steps shown in Fig. 3. With a negligible quality loss, **AT-EDM achieves 52.7%, 15.4%, 17.6% speed-up in terms of latency w/o FO, w/ FO under CI, w/ FO under DI, respectively**, which outperforms the state-of-the-art work by a clear margin. We present the memory footprint of AT-EDM in Supplementary Material.

4.4. Ablation Study

Self-Attention (SA) vs. Cross-Attention (CA). G-WPR can potentially use attention maps from self-attention (SA-based WPR) and cross-attention (CA-based WPR). We provide a detailed comparison between the two implementations. We visualize their pruning masks and provide gener-

Table 2. Comparison between sampling latency in different cases. † means not deploying pruning at the second feature level.

Model	SD-XL	ToMe	AT-EDM	AT-EDM†
Ave. FLOPs/step	6.7 T	4.1 T	4.1 T	4.5 T
w/o FO	31.0s	21.0s	**20.3s**	22.1s
w/ FO under CI	18.0s	17.7s	18.3s	**15.6s**
w/ FO under DI	18.0s	17.7s	16.3s	**15.3s**

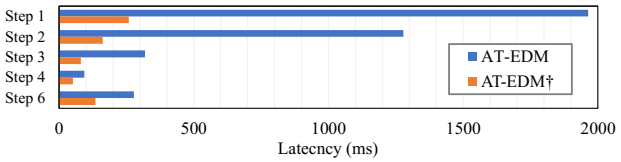


Figure 8. Latency incurred by different pruning steps shown in Fig. 3. Measured w/ FO under CI. Note that under DI, the latency of Step 1 (get the attention map) is eliminated.

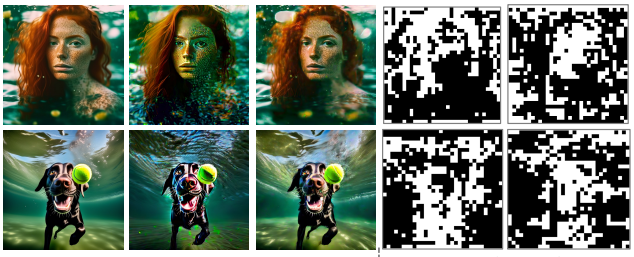


Figure 9. Comparison between different implementations of G-WPR: CA-based WPR and SA-based WPR. In general, CA-based WPR may remove too many background tokens, making the background not recoverable, while SA-based WPR preserves the image quality better.

ated image examples for a visual comparison in Fig. 9. This figure indicates that SA-based WPR outperforms CA-based WPR. The reason is that CA-based WPR prunes too many background tokens, making it hard to recover the background via similarity-based copy.

Similarity-based Copy. We provide comparisons between different methods to fill the pruned pixels in Fig. 10, which demonstrate the advantages of our similarity-based copy method. Images generated by bicubic interpolation are quite similar to those generated by padding zeros because interpolation usually assigns near-zero values to pruned tokens that are surrounded by other pruned tokens and can hardly recover them. Direct copy means directly copying corresponding token values before the first pruning layer in the attention block to recover the pruned tokens, where the following attention layers do not process the copied values. Thus, the copied values cannot recover the information in pruned tokens and even negatively affect the retained tokens. On the contrary, similarity-based copy uses attention maps and tokens that are retained to recover the pruned tokens, providing significantly higher image quality.

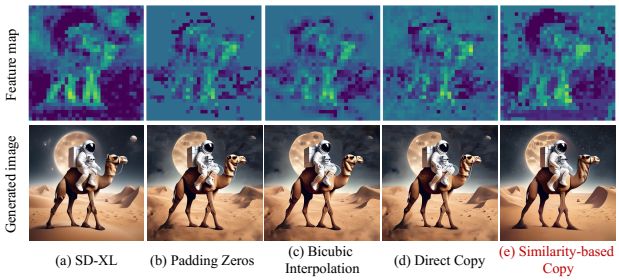


Figure 10. Different methods to recover the pruned tokens. Zero padding (**Col. b**), bicubic interpolation (**Col. c**), and direct copy (**Col. d**) can hardly recover pruned tokens and result in noticeable image degradation with blurry background (incomplete moon). On the other hand, similarity-based copy (**Col. e**) provides better image quality and keeps the complete moon in the original image. Better viewed when zoomed in.

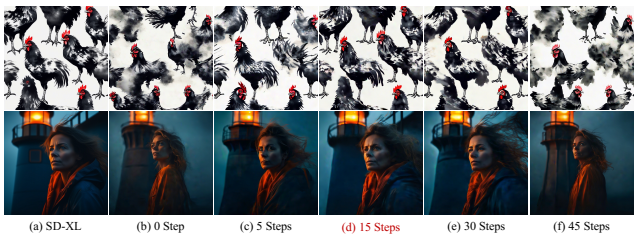


Figure 11. Comparison between different numbers of early prune-less steps where 0 step is the same as without DSAP. We find that pruning less on the first 15 steps achieves the best quality.

Denoising-Steps-Aware Pruning. We explore different design choices for DSAP.

(1) The prune-less schedule selects one attention block from each down-stage and up-stage in the U-Net and skips the token pruning in it. According to ablation results shown in Supplementary Material, F-L (First-Last) appears to be the best one, i.e., leaving the first attention block of down-stages and the last attention block of up-stages unpruned in early denoising steps.

(2) We then explore how the number of early prune-less denoising steps affects the generated image quality in Fig. 11. Note that we keep the FLOPs budget fixed and adjust the pruning rate accordingly when we change the number of prune-less steps. This figure shows that the setting of 15 early prune-less steps provides the best image quality. Note that the setting of zero prune-less step is identical to the setting without DSAP, and 5, 15, 30, 45 prune-less steps represents setting the boundary in Region I, II, III, IV of Fig. 5, respectively. The results indicate that placing the boundary between the prune-less and normal schedule in Region II performs best. This meets our expectation because the variance of attention maps becomes high enough to identify unimportant tokens well in Region II.

5. Conclusion

In this article, we proposed AT-EDM, a novel framework for accelerating DMs at run-time without retraining. AT-EDM has two components: a single-denoising-step token pruning algorithm and a cross-step pruning schedule (DSAP). In the single-denoising-step token pruning, AT-EDM exploits attention maps in pre-trained DMs to identify unimportant tokens and prunes them to accelerate the generation process. To make the pruned feature maps compatible with the latter convolutional blocks, AT-EDM again uses attention maps to reveal similarities between tokens and copies similar tokens to recover the pruned ones. DSAP further improves the generation quality of AT-EDM. We find such a pruning schedule can also be applied to other methods like ToMe. Experimental results demonstrate the superiority of AT-EDM with respect to image quality and text-image alignment compared to state-of-the-art methods. Specifically, on SD-XL, AT-EDM achieves a 38.8% FLOPs saving and up to 1.53$\times$ speed-up while obtaining nearly the same FID and CLIP scores as the full-size model, outperforming prior art.

Acknowledgment

This work was supported in part by an Adobe summer internship and in part by NSF under Grant No. CCF-2203399.

References

[1] Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4598–4602, 2023.

[2] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. *arXiv preprint arXiv:2210.09461*, 2022.

[3] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. MasaCtrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. *arXiv preprint arXiv:2304.08465*, 2023.

[4] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[5] Dave Epstein, Allan Jabri, Ben Poole, Alexei A Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. *arXiv preprint arXiv:2306.00986*, 2023.

[6] Gongfan Fang, Xinyin Ma, and Xinchao Wang. Structural pruning for diffusion models. *arXiv preprint arXiv:2305.10924*, 2023.

[7] Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In *Proceedings of the European Conference on Computer Vision*, pages 396–414. Springer, 2022.

[8] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[10] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[11] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1):2249–2281, 2022.

[12] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023.

[13] Bo-Kyeong Kim, Hyoung-Kyu Song, Thibault Castells, and Shinkook Choi. On architectural compression of text-to-image diffusion models. *arXiv preprint arXiv:2305.15798*, 2023.

[14] Woojeong Kim, Suhyun Kim, Mincheol Park, and Geunseok Jeon. Neuron merging: Compensating for pruned neu-

rons. *Advances in Neural Information Processing Systems*, 33:585–595, 2020.

[15] Woosuk Kwon, Sehoon Kim, Michael W Mahoney, Joseph Hassoun, Kurt Keutzer, and Amir Gholami. A fast post-training pruning framework for transformers. *Advances in Neural Information Processing Systems*, 35:24101–24116, 2022.

[16] Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, and Daniel Haziza. xFormers: A modular and hackable transformer modelling library. https://github.com/facebookresearch/xformers, 2022.

[17] Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. SRDiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022.

[18] Lijiang Li, Huixia Li, Xiawu Zheng, Jie Wu, Xuefeng Xiao, Rui Wang, Min Zheng, Xin Pan, Fei Chao, and Rongrong Ji. AutoDiffusion: Training-free optimization of time steps and architectures for automated diffusion model acceleration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7105–7114, 2023.

[19] Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. SnapFusion: Text-to-image diffusion model on mobile devices within two seconds. *arXiv preprint arXiv:2306.00980*, 2023.

[20] 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 *Proceedings of the European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[21] Enshu Liu, Xuefei Ning, Zinan Lin, Huazhong Yang, and Yu Wang. OMS-DPM: Optimizing the model schedule for diffusion probabilistic models. *arXiv preprint arXiv:2306.08860*, 2023.

[22] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022.

[23] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

[24] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik P. Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14297–14306, 2023.

[25] Or Patashnik, Daniel Garibi, Idan Azuri, Hadar Averbuch-Elor, and Daniel Cohen-Or. Localizing object-level shape variations with text-to-image diffusion models. *arXiv preprint arXiv:2303.11306*, 2023.

[26] 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. *arXiv preprint arXiv:2307.01952*, 2023.

[27] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[29] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[30] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[31] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[32] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[33] Suraj Srinivas and R. Venkatesh Babu. Data-free parameter pruning for deep neural networks. *arXiv preprint arXiv:1507.06149*, 2015.

[34] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

[35] Hongjie Wang, Bhishma Dedhia, and Niraj K. Jha. Zero-TPrune: Zero-shot token pruning through leveraging of the attention graph in pre-trained transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[36] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. *arXiv preprint arXiv:2112.07804*, 2021.

[37] Shuai Yang, Yukang Chen, Luozhou Wang, Shu Liu, and Yingcong Chen. Denoising diffusion step-aware models. *arXiv preprint arXiv:2310.03337*, 2023.

[38] Xingyi Yang, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Diffusion probabilistic model made slim. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22552–22562, 2023.

[39] Edouard Yvinec, Arnaud Dapogny, Matthieu Cord, and Kevin Bailly. Red: Looking for redundancies for data-free structured compression of deep neural networks. *Advances in Neural Information Processing Systems*, 34:20863–20873, 2021.

Attention-Driven Training-Free Efficiency Enhancement of Diffusion Models

Supplementary Material

The Supplementary Material is organized as follows. We first provide more implementation details of AT-EDM in Section A, including a detailed illustration of the SD-XL backbone. Then, we provide a more comprehensive comparison with the state-of-the-art method, ToMe [1], in Section B, including an analysis of why ToMe performs worse on SD-XL [26] than on previous versions of Stable Diffusion Models (SDMs). We provide more ablation results in Section C to justify our design choices in the main article. We analyze the memory footprint of AT-EDM in Section D. AT-EDM is orthogonal to various efficient DM methods, such as sampling distillation, thus can further boost their efficiency. To support this claim, we deploy AT-EDM in the distilled version of SD-XL, SDXL-Turbo[1], and show corresponding experimental results in Section E. We discuss limitations and trade-offs of AT-EDM in Section F and potential negative social impacts of AT-EDM in Section G.

A. Implementation Details

In this section, we provide more details of the implementation of AT-EDM. We first introduce the architecture of our SD-XL backbone as background material and then describe our single-step and cross-step pruning schedules in detail. We describe details of the evaluation and our calibration block for FLOPs measurement in the end.

A.1. The SD-XL Backbone

The state-of-the-art version of SDM is SD-XL. Compared with previous versions of SDM, it increases the quality of generated images significantly. Thus, we select SD-XL as the backbone model in this article. Specifically, we deploy AT-EDM and ToMe on `SDXL-base-0.9`. The architecture has two main differences from that of previous SDMs, such as SD-v1.5 and SD-v2.1: (1) attention blocks at the highest feature level (i.e., with the most tokens) are deleted; (2) attention blocks can potentially include multiple attention layers (an attention layer is composed of self-attention, cross-attention, and feed-forward network), such as A2 (includes 2 attention layers) and A10 (includes 10 attention layers).

To validate the conclusion that the cost of attention layers dominates the sampling cost, we investigate the FLOPs cost of SD-XL. Its FLOPs profile is shown in Fig. 12. This figure indicates that the attention block dominates the computational cost of all stages that include attention. We also investigate the scaling law of SD-XL at different generation resolutions, as shown in Fig. 13. We observe that the attention block dominates the cost at all resolutions. Note that the FLOPs cost of attention blocks does not scale much faster than that of ResNet blocks when the generation resolution increases. We believe this is due to the elimination of attention blocks at the highest feature level and the addition of attention layers at the lowest feature level, making the cost of feed-forward layers, which scales linearly with an increment in token numbers, a huge part of the cost of attention layers.

[1]https://huggingface.co/stabilityai/sd-turbo

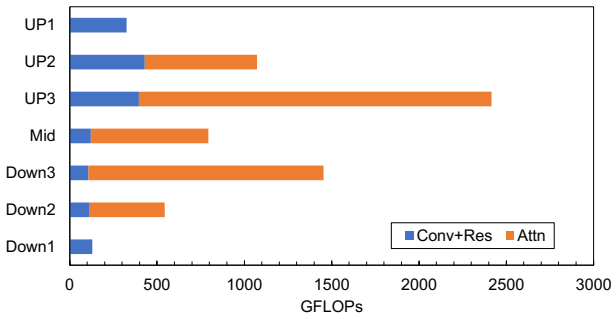


Figure 12. The FLOPs breakdown of SD-XL. Measured with 1024×1024 px image generation.

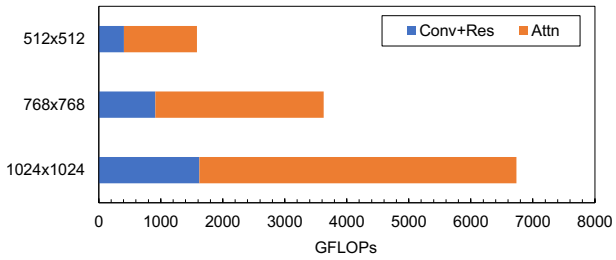


Figure 13. The FLOPs breakdown of ResNet blocks and attention blocks in SD-XL at different image resolutions.

A.2. Pruning in a Single Denoising Step

For a concise design, we always insert the pruning layer after the first attention layer of each attention block. All the other attention layers in this attention block can benefit from the reduction in token numbers. We may also insert multiple pruning layers at various locations in an attention block, which prunes tokens gradually. However, this requires a more thorough hyperparameter search to ensure a good balance between FLOPs cost and image quality.

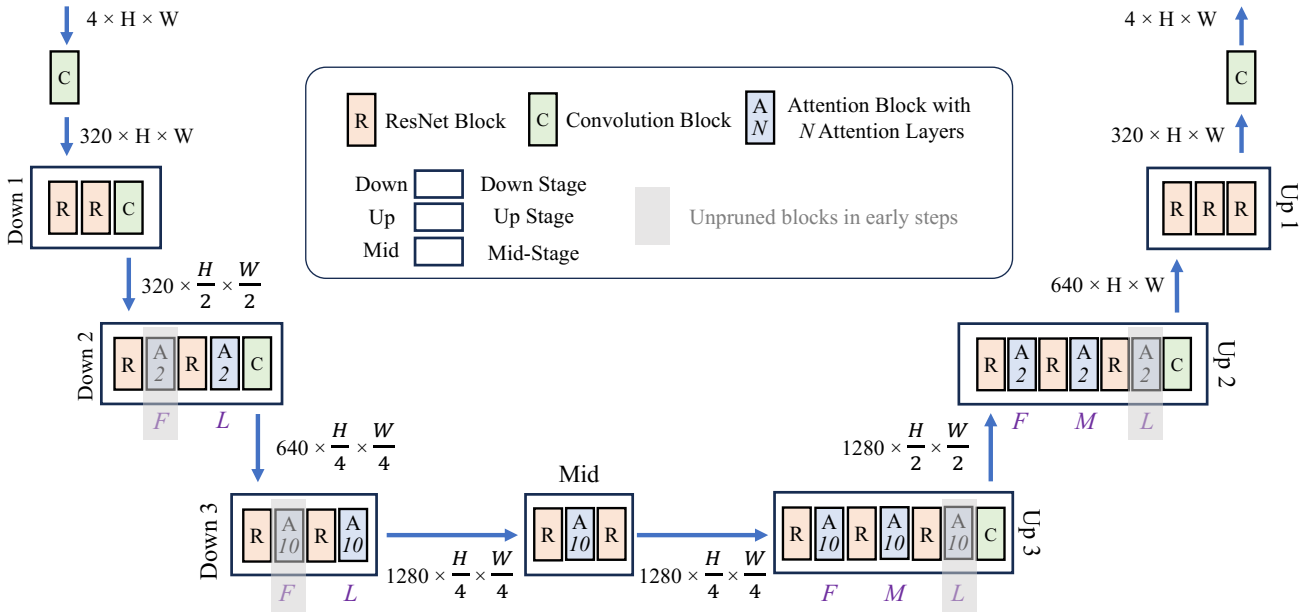


Figure 14. The U-Net architecture of SD-XL. Residual connections are not shown here for brevity. The example in this figure generates a $8H \times 8W$ pixel image. The input/output size of each stage is shown in the $C \times H \times W$ format, where C is the number of channels; H and W represent the resolution. There are two attention blocks {F(First), L(Last)} in each downsampling stage and three {F(First), M(Middle), L(Last)} in each upsampling stage. In the prune-less schedule, we do not apply pruning to attention blocks in the gray rectangles. Downsampling stage 1, 2, and 3 is at the first, second, and third feature level, respectively. AT-EDM† does not apply pruning to attention blocks at the second feature level.

A.3. The Prune-Less Schedule

Early denoising steps determine the layout of the generated images and have a weaker ability to differentiate between unimportant tokens [8]. Thus, we need heterogeneous denoising steps and, hence, use a less aggressive pruning schedule for some of the early denoising steps.

In the normal pruning setting, when we target 4.1 TFLOPs for each sampling step, we use a pruning rate of 63% (i.e., retain 37% tokens) after the first attention layer of A2 and A10; in the prune-less schedule, we do not apply pruning to attention blocks in the gray rectangles shown in Fig. 14. We validate the choice of not deploying pruning through ablative experimental results shown in the main article.

A.4. Details of Evaluation

When measuring the FID and CLIP scores on MS-COCO 2017 [20], we deduplicate captions to make sure each image corresponds to a single caption. We center cropped images in the validation set, resize them to 1024×1024 px, and use the `clean-fid` library[2] to calculate FID scores. We use the ViT-G/14 model of Open-CLIP[3] to calculate the CLIP scores of generated images. We set the batch size to 3 when we generate images for visual comparison and quantitative analysis. We run all experiments on a single NVIDIA A100-40GB GPU.

A.5. Calibration Block for FLOPs Measurement

The popular library for FLOPs measurement, `fvcore`[4], is not natively compatible with SDMs. Thus, we use the `THOP`[5] library instead to measure the FLOPs cost of SDMs. However, we found it does not correctly compute the FLOPs cost of self-attention. The FLOPs cost of sampling steps given by this library scales linearly as the number of image tokens. This is unreasonable because the cost of self-attention in sampling steps scales quadratically when the number of tokens increases (other parts of a sampling step scale linearly). After a thorough investigation of the behavior of `THOP`, we found it basically does not take the cost of self-attention into account. Thus, we design a calibration block to supplement the missed term of FLOPs cost for each attention block:

$$F_{cali} = 4 \times B \times N_a \times (HW)^2 \times C \qquad (2)$$

where B is the batch size; N_a is the number of attention layers in this attention block; HW is the number of image

[2]https://github.com/GaParmar/clean-fid/tree/main
[3]https://github.com/mlfoundations/open_clip
[4]https://github.com/facebookresearch/fvcore
[5]https://github.com/Lyken17/pytorch-OpCounter

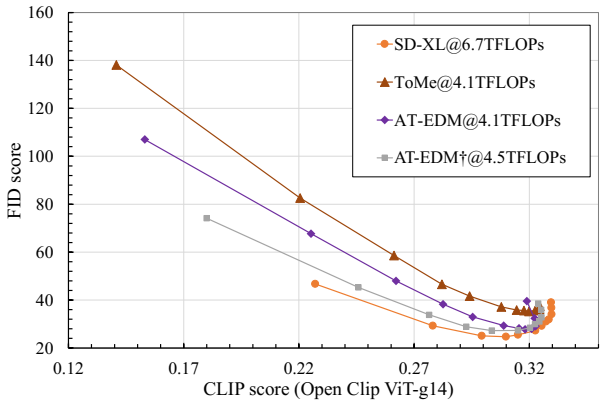


Figure 15. Complete FID-CLIP score curves. The used CFG scales are [1.0, 1.5, 2.0, 2.5, 3.0, 4.0, 5.0, 6.0, 7.0, 9.0, 12.0, 15.0].

tokens; and C is the number of channels. The factor 4 is due to the fact that (1) there are two images processed at the same time for each generated image in a batch (one is guided by the prompt, and another is not); (2) there are two Matrix-Matrix Multiplications (MMMs) in self-attention.

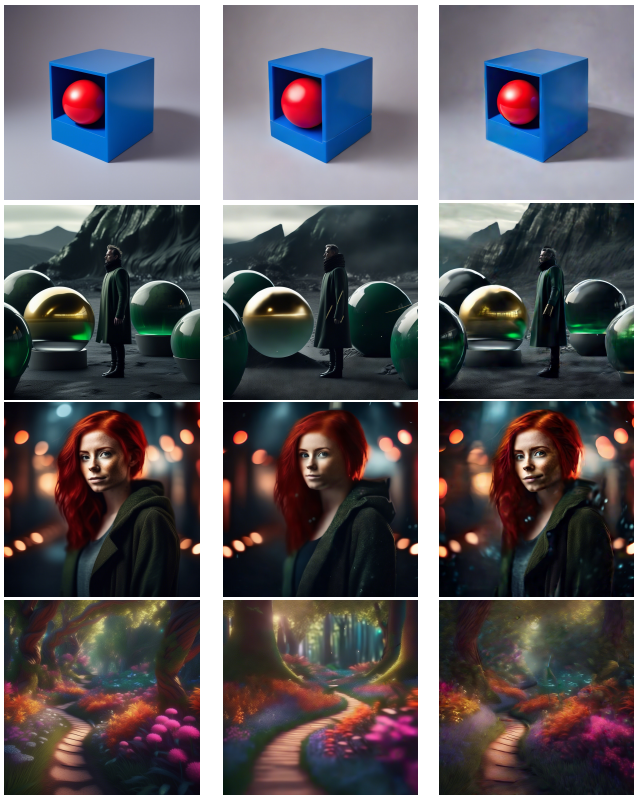
B. Comprehensive Comparison with ToMe

In this section, we first analyze why ToMe cannot replicate on SD-XL its good performance on previous SDMs in Section B.1. Then, we provide complete FID-CLIP curves to compare AT-EDM with ToMe in Section B.2. In the end, we present cases in which both AT-EDM and ToMe perform well and visually compare AT-EDM and ToMe under various FLOPs budgets in Section B.3.

B.1. Deploying ToMe on SD-XL

For SD-v1.x and SD-v2.x, ToMe maintains the generated image quality quite well after token merging. However, as we demonstrate in the main article, ToMe incurs obvious quality degradation on SD-XL after token merging.

In the default setting of ToMe, it only merges tokens for attention blocks at the highest feature level and their self-attention. However, SD-XL eliminates attention blocks at the highest abstraction level and native ToMe does not do anything to this backbone. Thus, it is necessary to expand its merging range to **attention blocks at all feature levels**. In addition, since SD-XL adds a lot more attention layers at the lowest feature level, where tokens are significantly fewer than at higher feature levels, self-attention no longer dominates the cost of attention layers. Given that the merging ratio of ToMe has an upper limit of 75%, it is not enough to only merge tokens for self-attention to meet the 4.1 TFLOPs budget. Thus, it is necessary to expand its merging range to **Cross-Attention (CA), Self-Attention (SA), and the Feed-Forward (FF) network**. We believe the expanded deployment range of token merging leads to the relatively poor performance of ToMe on SD-XL. Note



(a) SD-XL (b) ToMe (c) AT-EDM

Figure 16. Examples on which both AT-EDM and ToMe perform well. Each row of this figure corresponds to the following typical cases: (1) simple single main object with a simple background; (2) multiple main objects; (3) complex single main object; (4) complex scene without a main object.

that to meet the 4.1 TFLOPs budget for each sampling step, we set the merging ratio to 50% for ToMe under the expanded merging range.

B.2. Complete FID-CLIP Curves

We explore the trade-off between the CLIP and FID scores through various CFG scales. We show the complete FID-CLIP curves in Fig. 15. AT-EDM† does not deploy pruning at the second feature level (as mentioned in the caption of Fig. 14). This figure illustrates that for most CFG scales, AT-EDM not only lowers the FID score but also results in higher CLIP scores than ToMe, implying that images generated by AT-EDM not only have better quality but also better text-image alignment.

B.3. More Images from AT-EDM and ToMe

In some cases, ToMe performs fairly well and has its merits. We present several typical examples in Fig. 16. The first example in the first row represents the case of a simple main object with a simple background. Both ToMe and AT-EDM

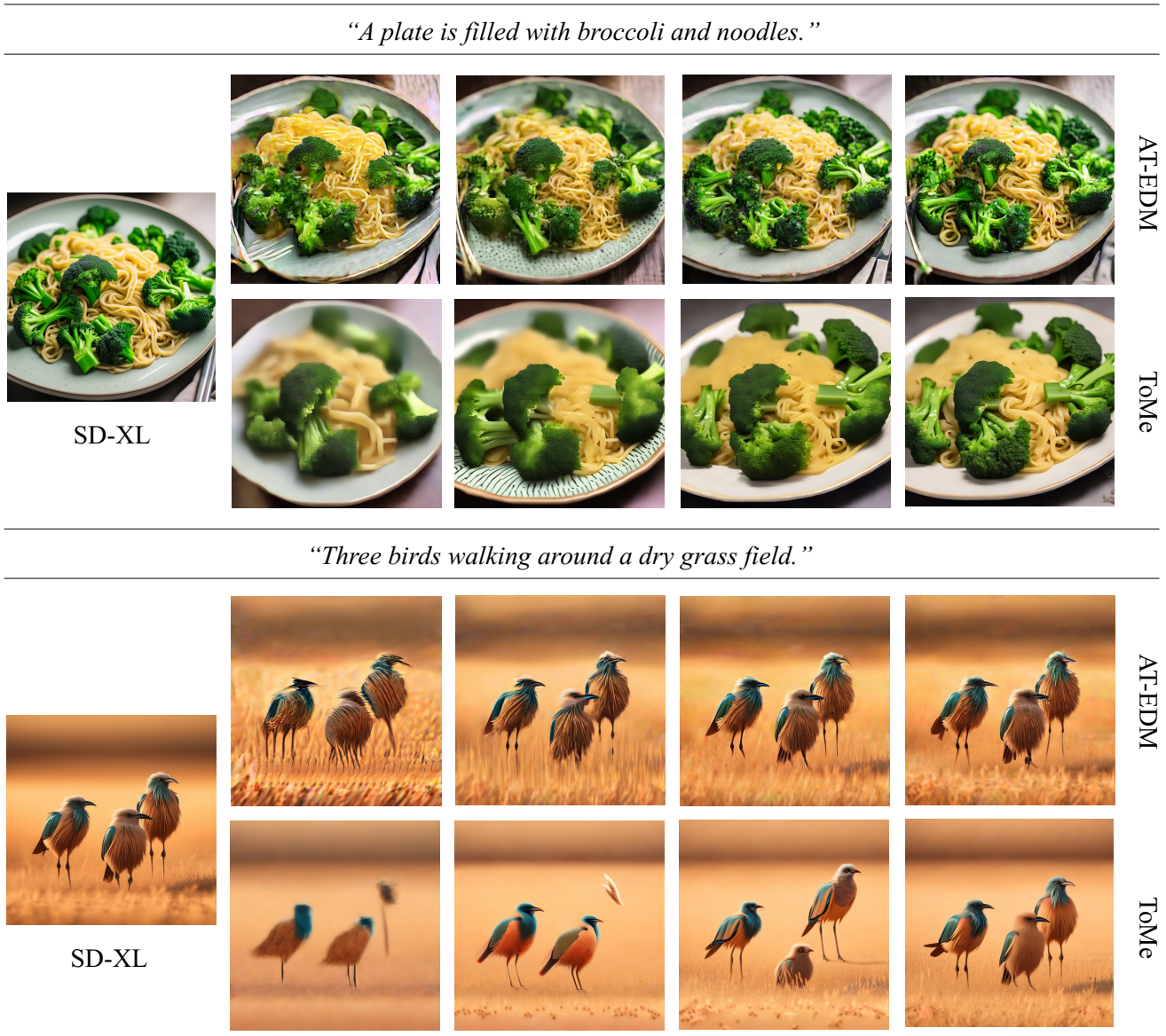


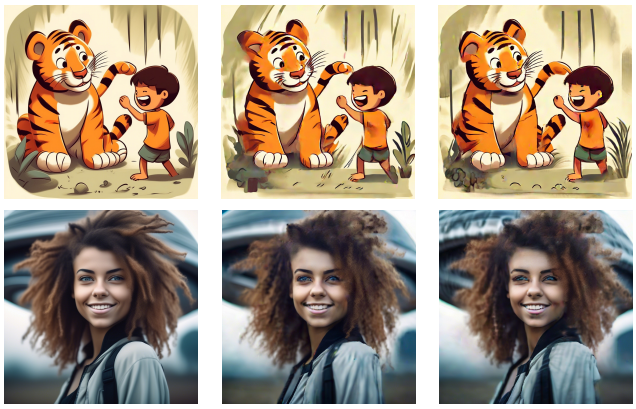
Figure 17. Comparison between AT-EDM and ToMe under different FLOPs budgets. Note that for **Col.e**, the average cost of each sampling step for AT-EDM (ToMe) is 4.52 (4.56) TFLOPs. Prompts are selected from the MS-COCO 2017 validation dataset.

preserve the main object quite well. The second row represents a more complex case in which there are multiple main objects in the generated image. Although ToMe loses some texture details, it preserves the overall layout quite well. The third row is the case of a typical complex main object, a human face. In this example, ToMe preserves the face without artifacts. The last row of this figure demonstrates the case of generating a complex scene without a main object. In this case, both ToMe and AT-EDM can maintain the layout well while supplementing some details. These examples show that ToMe is a strong baseline and it is non-trivial to outperform it.

We also provide visual examples of ToMe and AT-EDM under different FLOPs budgets in Fig. 17. It indicates that AT-EDM outperforms ToMe under any FLOPs budget. We also observe that AT-EDM needs at least 3.6 TFLOPs budget to ensure an acceptable image quality.

C. More Ablation Experiments

In this section, we supplement ablation experiments to validate our design choices. We first discuss the deployment



(a) SD-XL (b) After-FF (c) Before-FF

Figure 18. Comparison between inserting the pruning layer after the FF and before the FF layer.

location for run-time pruning and then compare different implementations of the mapping function $f(\mathbf{A}, s_K)$ for CA-based WPR. Note that CA-based WPR and SA-based WPR are two implementations of G-WPR and we mainly focus on CA-based WPR in this section. We also investigate the schedule that prunes more in early denoising steps and verify our intuition of pruning less in early steps.

C.1. Deployment Location for Run-Time Pruning

In our default setting, we use generated masks after the FF layer to perform token pruning. Another option is to perform pruning early before the FF layers, which results in a little bit of extra FLOP savings at the cost of image quality. We provide several visual examples in Fig. 18. Note that here, we simply change the pruning layer insertion location without keeping the total FLOPs cost fixed, which is different from what we do in the ablation experiments in the main article. We find that inserting the pruning layer before the FF layer indeed hurts image quality (although slightly). For example, the plant in the first example and the UFO in the second example become worse. Given that pruning before the FF layer only results in marginal extra FLOPs savings (reduces the cost from 4.1 TFLOPs to 4.0 TFLOPs), we choose to prune after the FF layer to obtain better image quality.

C.2. Implementations of CA-based WPR

To generalize WPR to cross-attention, we need to design a function $f(\mathbf{A}, s_K)$ that maps the importance of Key tokens to that of Query tokens. The intuition behind designing this function is that vital Query tokens should devote much of their attention to important Key tokens. Thus, the desired attention distribution should satisfy: (1) similarity to the importance distribution of Key tokens; (2) concentration on a few tokens. Then, when designing $f(\mathbf{A}, s_K)$, we need to (1) reward the similarity between the attention distribution (i.e., each row of $\mathbf{A}$) and the importance distribution (i.e., s_K); (2) penalize uniform attention distribution. Based on these points, we obtain several implementations of $f(\mathbf{A}, s_K)$. We had mentioned an entropy-based implementation in the main article, which rewards similarity through the dot-product and penalizes uniform distribution through entropy. We provide additional implementations here:
(I) **Hard-clip**-based implementation

$$s_Q^{t+1}(x_i) = f(\mathbf{A}, s_K^{t+1}) = \sum_{j=1}^{N} \epsilon(A_{i,j} - \eta) \cdot s_K^{t+1}(x_j) \quad (3)$$

where $\epsilon(x) = 1$ if $x \geq 0$, $\epsilon(x) = 0$ if $x < 0$; η is the threshold of attention (we set it to 0.2 as the default setting); $A_{i,j}$ is the attention from Query q_i to Key k_j.
(II) **Soft-clip**-based implementation

$$s_Q^{t+1}(x_i) = f(\mathbf{A}, s_K^{t+1}) = \sum_{j=1}^{N} \text{Sig}(A_{i,j} - \eta) \cdot s_K^{t+1}(x_j) \quad (4)$$

where $\text{Sig}(x) = \frac{1}{1+e^{-x}}$.

(III) **Power**-based implementation

$$s_Q^{t+1}(x_i) = f(\mathbf{A}, s_K^{t+1}) = \sum_{j=1}^{N} (\beta \cdot s_K^{t+1}(x_j))^{\alpha \cdot A_{i,j}} \quad (5)$$

where α and β are scaling factors to ensure that $\beta \cdot s_K^{t+1}(x_j) > 1$ and $\alpha \cdot A_{i,j} > 1$ for large $s_K^{t+1}(x_j)$ and $A_{i,j}$. Here, we let $\alpha = 5$ and $\beta = \frac{N_t}{2}$, where N_t denotes the number of Key tokens.

We compare these implementations visually in Fig. 19. We find that among these implementations, the hard-clip-based implementation performs the worst. Although the entropy-based implementation and the power-based implementation are better than other implementations for CA-based WPR, none of them can outperform SA-based WPR. Thus, we use SA-based WPR as our default setting in AT-EDM.

C.3. Prune-Less Schedule for Early Denoising Steps

The prune-less schedule selects one attention block from each down-stage and up-stage in the U-Net and skips the token pruning in it. We generate images with the same prompts and different selections, as shown in Fig. 20. It indicates that F-L appears to be the best choice. F-L is the schedule that we show in Fig. 14.

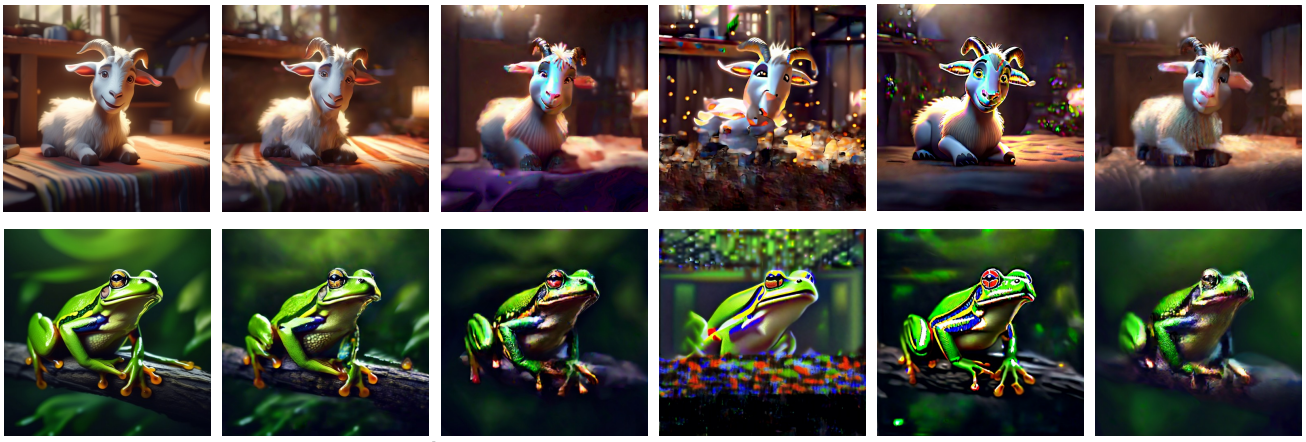


Figure 19. Comparison between different implementations of Cross-Attention-based WPR. None of them can outperform Self-Attention-based WPR.

Figure 20. Comparison between different prune-less settings. There are two attention blocks {F(First), L(Last)} that are left unpruned in the downsampling stages and three {F(First), M(Middle), L(Last)} in the upsampling stages. Results indicate that F-L is the best schedule.

C.4. The Number of Prune-Less Steps

The intuitions that we use to design the prune-less schedule in the early denoising steps are (1) early denoising steps determine the layout of generated images and thus are crucial; (2) early denoising steps have a weaker ability to differentiate unimportant tokens. The first intuition prohibits us from pruning more tokens in the early steps (see Section C.5). The second intuition guides us to choose the number of prune-less steps. The variance of attention maps reflects their ability to differentiate unimportant tokens since the attention score of unimportant tokens deviates significantly from that of normal tokens. We show the variance of attention maps given by different denoising steps in Fig. 5. The figure indicates that the variance is more than 1.0E-5 after the first 15 denoising steps. This supports our hyperparameter choice.

C.5. Prune More in Early Denoising Steps

In AT-EDM, we design a cross-step pruning schedule that is less aggressive in early denoising steps. This is based on the intuition that (1) early denoising steps determine the layout of generated images and thus are very important; (2) the ability of early denoising steps to differentiate between unimportant tokens is weaker than that of later steps. To verify our intuition, we investigate the schedule that prunes more in early denoising steps. Note that for symmetry, *"prune more in the first 15 steps"* selects corresponding attention blocks in the last 35 steps for not pruning tokens

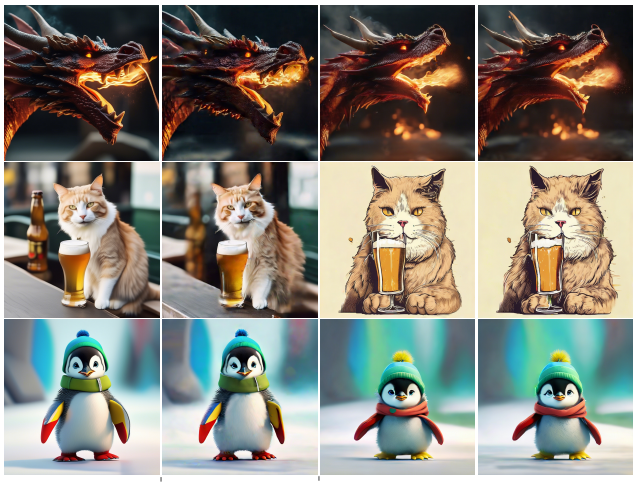


Figure 21. Comparison between different DSAP schedules. Examples indicate that pruning more tokens in early denoising steps changes the layout of generated images significantly.

while keeping the total FLOPs cost fixed. We provide visual examples in Fig. 21 for comparison. These examples clearly support our intuition that pruning more in early denoising steps not only affects the layout of generated images but also hurts image quality.

D. Memory Footprint of AT-EDM

Since we need to obtain the attention map from the first attention layer, AT-EDM cannot reduce the peak memory footprint. However, benefiting from the significantly reduced number of tokens in the following attention layers, AT-EDM reduces the average memory footprint significantly. Since PyTorch does not automatically release the redundant assigned memory when the memory requirement reduces in the later layers, we theoretically estimate the average footprint of AT-EDM, assuming the redundant occupied memory will be released in the layers with fewer tokens. We believe this is practical when the implementation is good enough. The peak and theoretical average footprint of full-size SD-XL (AT-EDM) are 19.5GB (19.5GB) and 18.8GB (12.6GB), respectively. This indicates that if we have a fine-grained pipeline schedule, **AT-EDM allows us to run 49.2% more generation tasks** with the given VRAM restriction.

E. Stack with Sampling Distillation

Methods like consistency distillation [23, 32] can greatly reduce the cost of DMs. Note that AT-EDM does not contradict these methods and can be deployed to speed them up further. To support this, we deploy AT-EDM in SDXL-Turbo, which is a distilled version of SD-XL. Our experimental results show that although SDXL-Turbo reduces around 95% FLOPs cost of SD-XL, **AT-EDM can further reduce the FLOPs cost of SDXL-Turbo by 33.4% while reducing FID by 14.5% and only incurring 2.1% CLIP reduction** on MSCOCO-2017. AT-EDM works as a regularizer and slightly improves the quality of images.

F. Limitations and Trade-Offs

AT-EDM demonstrates state-of-the-art results for accelerating DM inference at run-time without any retraining. However, as a machine learning algorithm, it inevitably has some limitations.

(1) AT-EDM requires a pre-trained DM; since it saves computation to accelerate the model, its performance is inherently upper-bounded by the full-sized one. While most of the time, AT-EDM matches the performance of the pre-trained model, both quantitatively and qualitatively (see experimental results in the main article), with around 40% FLOPs reduction, there exist some samples where the full-sized model outperforms AT-EDM (see Fig. 17). Nonetheless, AT-EDM outperforms prior art by a clear margin. In addition, AT-EDM is differentiable. We will fine-tune the pruned model to further improve quality in the future.

(2) AT-EDM leverages the rich information stored in the attention maps, which could be inaccessible without incurring overhead due to the open-sourced nature of the implementation. For instance, SD-XL [26] adopts an efficient attention library, xFormers [16], which fuses computation to directly obtain succeeding tokens without providing intermediate attention maps. As shown in Table 2, in the case that Fused Operation (FO) is not used, using AT-EDM leads to significant latency savings. With FO under the Current Implementation (CI), AT-EDM does not result in a huge latency saving due to the cost of calculating attention maps. Reusing attention maps across steps and obtaining an approximation for them could alleviate this issue. With FO under the Desired Implementation (DI) that provides attention maps, AT-EDM's potential is fully unlocked and leads to a decent speedup.

AT-EDM is especially good at generating object-centric images, such as a portrait. It can employ a high pruning rate without hurting the main object. Generating complex scenes or tens of objects is relatively tricky for AT-EDM since it may lose some details in corner cases. In some rare corner cases where the texture details are not significant, ToMe might perform slightly better, as our algorithm may prune too many tokens in that small region. ToMe is indeed a strong baseline, but it is remarkable that our AT-EDM still outperforms it in most cases.

G. Potential Negative Social Impacts

Text-to-image generative models like SD-XL have significantly advanced the field of AI and digital art creation. However, they may also potentially have negative social impact. For example, they can create highly realistic images that may be indistinguishable from real photographs. As the technology can be used to create convincing but false images, this can potentially lead to confusion and misinformation spread. In addition, the use of these models to create inappropriate or harmful content, such as realistic images of violence, hate speech, or explicit material, raises significant ethical questions. There is also the potential for perpetuating biases if the AI model is trained on biased datasets.