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 [6] 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. Then we provide details of the evaluation and our calibration block for FLOPs measurement. We demonstrate the baseline methods of similarity-based copy in detail and provide the extra latency incurred by different pruning steps in the end.

A.1. The SD-XL Backbone

The state-of-the-art version of SDM is SD-XL. Compared with previous SDM versions [7], 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. 1. This

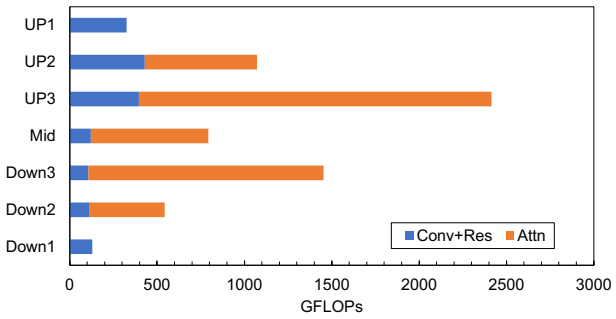


Figure 1. The FLOPs breakdown of SD-XL. Measured with 1024×1024 px image generation.

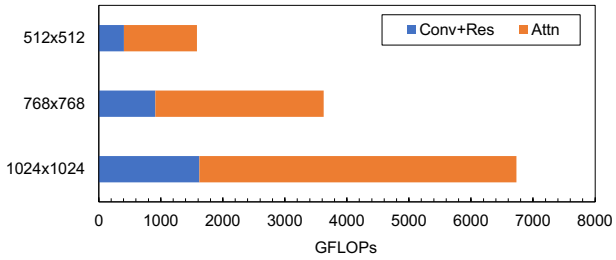


Figure 2. The FLOPs of ResNet blocks and attention blocks in SD-XL at different image resolutions.

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. 2. 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.

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 re-

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

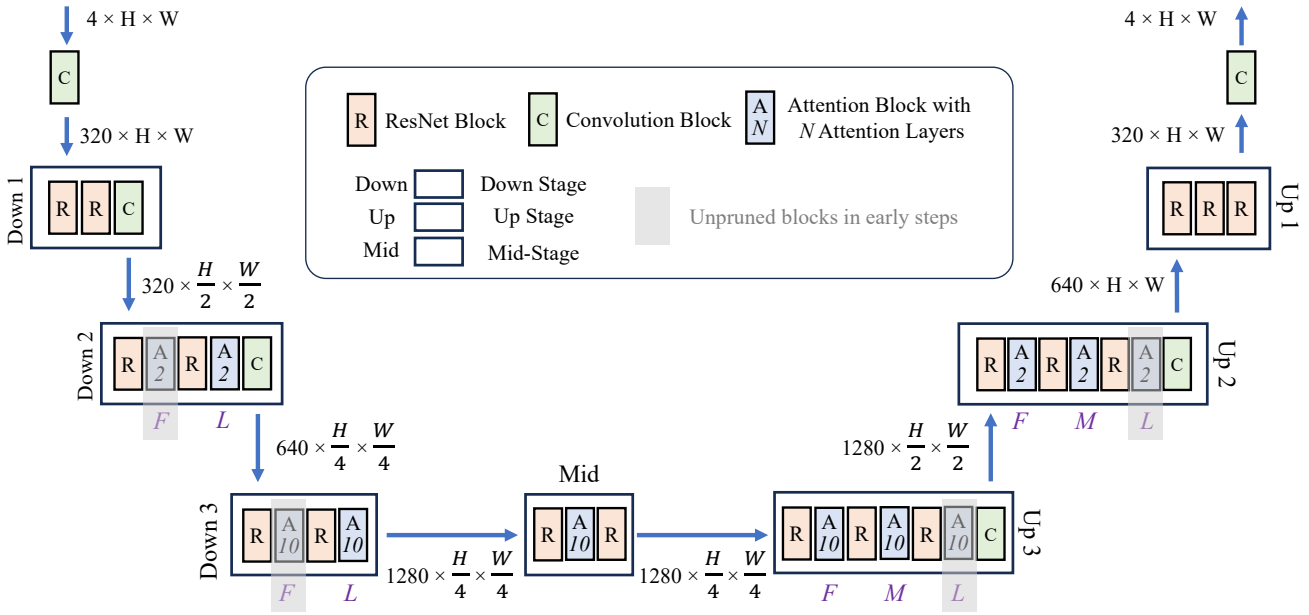


Figure 3. The U-Net architecture of SD-XL. Residual connections are not shown here for brevity. The example in this figure generates a $8H \times 8W$ pixel image. The input/output size of each stage is shown in the $C \times H \times W$ format, where C is the number of channels; H and W represent the 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.

quires a more thorough hyperparameter search to ensure a good balance between FLOPs cost and image quality.

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 [2]. 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. 3. 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 [4], 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 that 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:

[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

$$F_{cali} = 4 \times B \times N_a \times (HW)^2 \times C \quad (1)$$

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

A.6. Baselines of Similarity-based Copy

Here, we introduce several straightforward methods as the baselines of similarity-based copy to recover pruned tokens.
(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.
(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.

A.7. Extra Latency Incurred by Pruning

We measure the generation latency on a single A100 GPU. The deployment of pruning incurs extra latency. We show corresponding results in Fig. 4. This figure explains why AT-EDM† is even faster than AT-EDM with FO under CI, although AT-EDM† prunes less than AT-EDM (it does not perform pruning at the second feature level).

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 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.

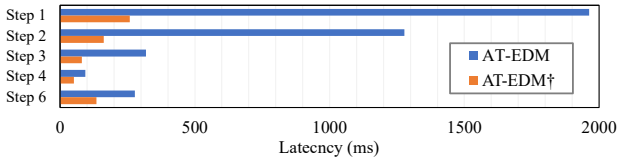


Figure 4. Latency incurred by different pruning steps shown in Fig. 3 of our main paper. Measured w/ FO under CI. Note that under DI, the latency of Step 1 (get the attention map) is eliminated.

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 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. 5. AT-EDM† does not deploy pruning at the second feature level (as mentioned in the caption of Fig. 3). 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 show several typical examples in Fig. 6. The first example in the first row represents the case of a simple main object with a simple background. Both ToMe and AT-EDM

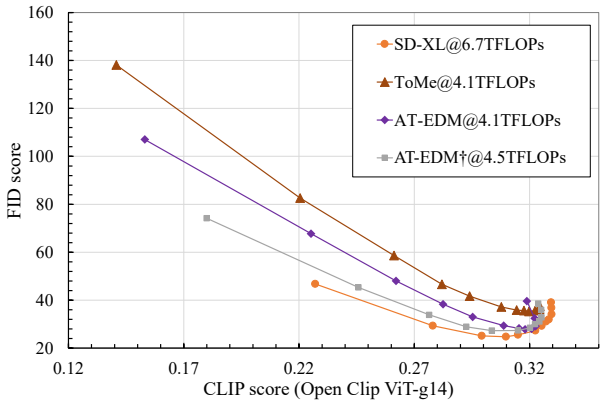


Figure 5. Complete FID-CLIP score curves. The used CFG scales are [1.0, 1.5, 2.0, 2.5, 3.0, 4.0, 5.0, 6.0, 7.0, 9.0, 12.0, 15.0].

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.

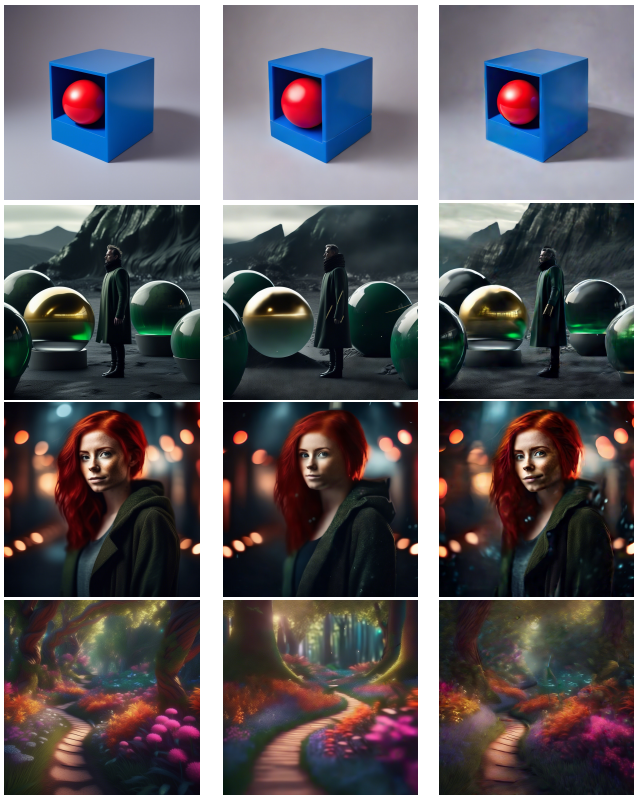
We also provide visual examples of ToMe and AT-EDM under different FLOPs budgets in Fig. 7. It indicates that AT-EDM outperforms ToMe under all FLOPs budgets. We also observe that AT-EDM needs at least a 3.6 TFLOPs budget to ensure an acceptable image quality.

C. More Ablation Experiments

In this section, we supplement ablation experiments to validate our design choices. We first discuss the deployment 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 FLOPs savings at the cost of image quality. We provide several visual examples in Fig. 8. Note that,



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

Figure 6. 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.

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 [9] 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) concen-

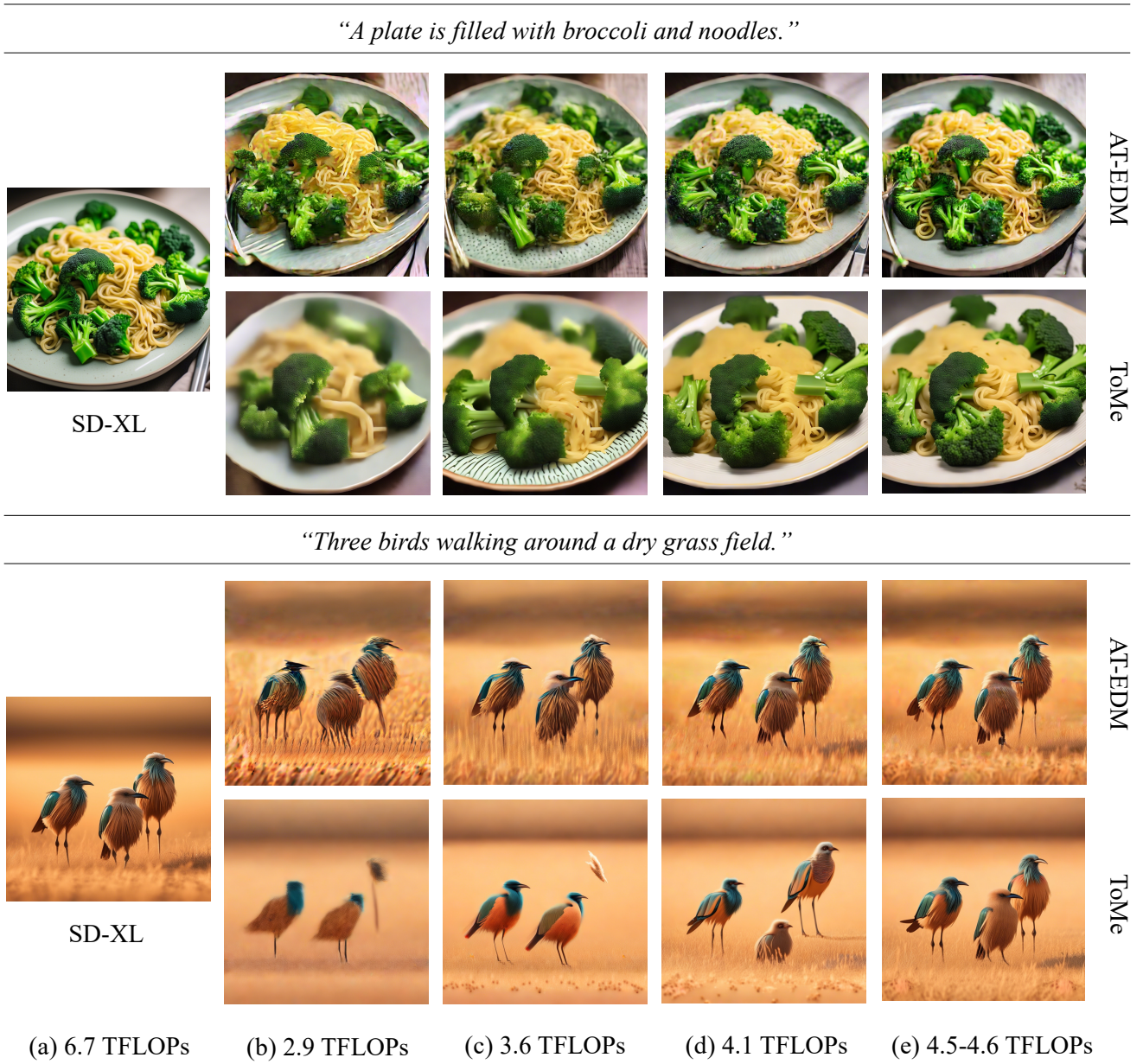


Figure 7. Comparison between AT-EDM and ToMe under different FLOPs budgets. Note that for **Col.e**, the average cost of each sampling step for AT-EDM (ToMe) is 4.52 (4.56) TFLOPs. Prompts are selected from the MS-COCO 2017 validation dataset.

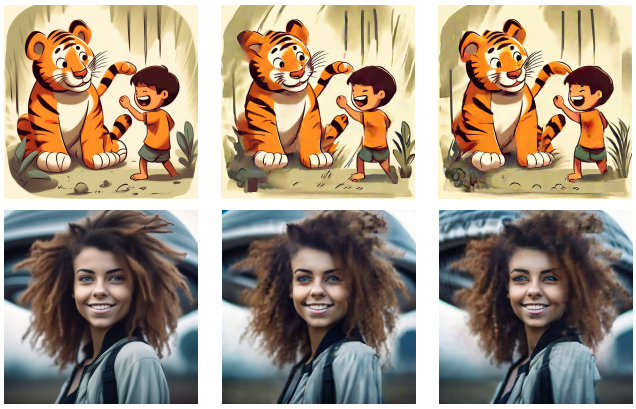
tration on a few tokens. Then, when designing $f(\mathbf{A}, s_K)$, we need to (1) reward the similarity between the attention distribution (i.e., each row of $\mathbf{A}$) and the importance distribution (i.e., s_K); (2) penalize uniform attention distribution. Based on these points, we obtain several 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

$$s_Q^{t+1}(x_i) = f(\mathbf{A}, s_K^{t+1}) = \sum_{j=1}^{N} \epsilon(A_{i,j} - \eta) \cdot s_K^{t+1}(x_j) \quad (2)$$

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



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

Figure 8. Comparison between inserting the pruning layer after the FF and before the FF layer.

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

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

(III) **Power**-based implementation

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

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. 9. 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. 10. It indicates that F-L appears to be the best choice. F-L is the schedule that we show in Fig. 3.

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 of our main paper. 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 while keeping the total FLOPs cost fixed. We provide visual examples in Fig. 11 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.

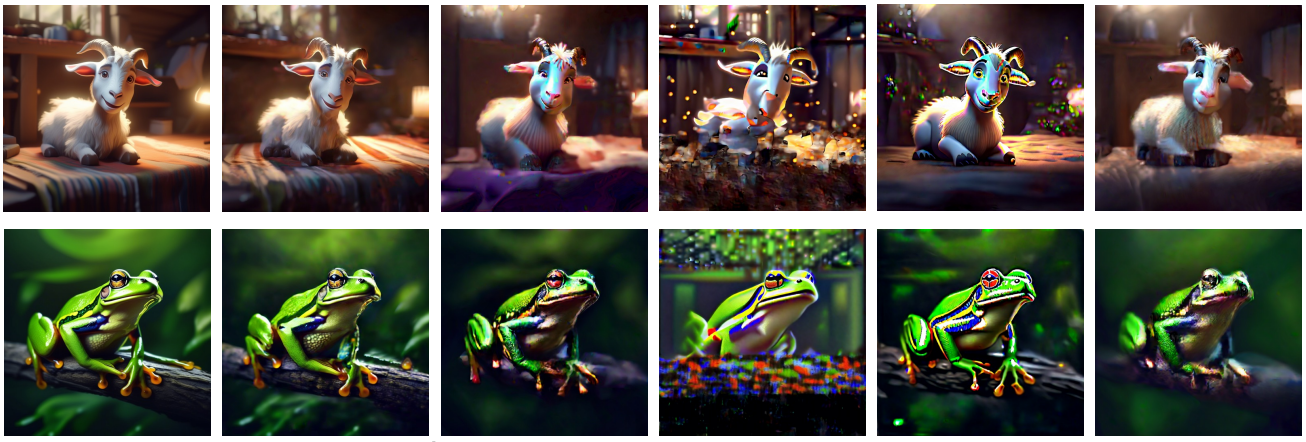


Figure 9. Comparison between different implementations of Cross-Attention-based WPR. None of them can outperform Self-Attention-based WPR.

Figure 10. 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.

E. Stack with Sampling Distillation

Methods like consistency distillation [5, 8] can greatly reduce the cost of DMs. Note that AT-EDM is compatible with these methods and can be deployed to speed them up further. To support this claim, 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 validation set. 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. 7). 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 [6] adopts an efficient attention library, xFormers [3], which fuses computation to directly

Figure 11. Comparison between different heterogeneous sampling schedules. Examples indicate that pruning more tokens in early denoising steps changes the layout of generated images significantly.

obtain succeeding tokens without providing intermediate attention maps. As shown in Table 2 of our main paper, 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 AT-EDM still outperforms it in most cases.

G. Potential Negative Social Impacts

Text-to-image generative models like SD-XL have brought significant advancements in 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.

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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.