Accelerating Conditional Image Generation with *CachedAttention*

Yi Pan*
conlesspan@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Ziyi Xu*
xzy2022@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Shihan Fang*
fang-account@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Prompt: *A bustling urban street scene with a red articulated bus prominently in the foreground.*

Figure 1: We propose *CachedAttention*, a training-free attention mechanism that adopts a dynamic cache-and-reuse policy to alleviate condition redundancy and speed up conditional image generation. It achieves up to 34% speedup with negligible loss in image quality.

ABSTRACT

Conditional image generation involves generating images based on specific input conditions, such as text or other modalities. Diffusion Transformers (DiT) are widely used in this task, leveraging cross-attention or unified self-attention to align generated images with conditions. While effective, these mechanisms can lead to substantial computational complexity, particularly when dealing with long prompts or complex conditions. We identify notable condition redundancy in this process, as attention outputs between image and condition tend to be similar across different timesteps. To address this issue, we propose *CachedAttention*, a training-free attention mechanism for accelerating DiTs. By recognizing the varying similarity of different timesteps and tokens in the condition, *CachedAttention* dynamically caches and reuses the intermediate results in attention operators, thereby reducing redundant computations. We integrate *CachedAttention* into PixArt-alpha and OmniGen, two popular image generation models. Evaluation results show that our method improves overall throughput by 34%, effectively mitigating condition redundancy in DiTs and improving the computational efficiency of conditional image generation.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; **Machine learning**; **Distributed computing methodologies**.

*All authors contributed equally to this research.

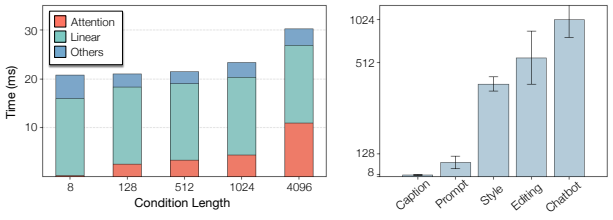
KEYWORDS

Image Generation; Diffusion Models; CachedAttention

1 INTRODUCTION

In recent years, diffusion transformers (DiT) [27] have gained increasing popularity in conditional image generation tasks, including text to image generation [8, 9], multi-modal to image generation [42, 48], and image editing [41]. Many of these works adopt cross-attention [9, 39] or unified multi-modal self-attention [41, 42, 48] to align the generated images with input conditions. While these mechanisms effectively follow instructions and generate high-quality images, they both share a common inefficiency of the attention operator: the computational complexity increases significantly with the length of the input condition [44]. As demonstrated in Figure 2, the attention operator becomes a primary bottleneck of the generation process as the input token length increases. Therefore, in some recent tasks like image editing [28] and multi-modal chatbot [4], where the length of input conditions can be up to 1000 tokens, the image generation throughput can be extremely low.

Previous efforts to accelerate transformer model inference have mainly focused on modifying the model architecture, such as grouped-query attention (GQA) [5] and multi-head latent attention (MLA) [26], which requires additional training costs. Some recent works [22, 43] have discovered temporal redundancy of diffusion models, where the outputs of neighboring timesteps have high similarity. Based on this, those works proposed optimizations to accelerate



(a) Latency breakdown (b) Condition length of common tasks

Figure 2: Latency breakdown of conditional image generation with different condition lengths.

the inference of diffusion models through asynchronous communication and attention compression. However, these methods have not considered the inefficiency of DiT when the input condition is significantly long.

In this work, we identify *condition redundancy* of DiT inference, and aim to address the previous inefficiency by reducing this redundancy. Specifically, we observe that the attention score and output between the output image and the input condition are similar across neighboring timesteps.

One possible solution to mitigate this problem is to cache and reuse the attention output across multiple timesteps. For instance, the attention output from timestep 1 could be reused for timesteps 1 through 5. However, this approach does not account for the varying levels of similarity between different timesteps and tokens in the input. For example, the attention outputs for timesteps 19 and 20 are often more similar than those for timesteps 1 and 2. Similarly, the attention scores between an image and unimportant tokens (e.g., `of`) may exhibit more similarity across timesteps compared to the scores between the image and more critical tokens (e.g., `cat`). Therefore, neglecting the varying similarity and treating all the timesteps and tokens together brings a significant tradeoff between the quality and speed of the generation process.

To tame this tradeoff, we propose *CachedAttention*, a novel approach to dynamically cache and reuse attention outputs at each timestep and for each block of input tokens. Unlike traditional static caching algorithms, *CachedAttention* adaptively decides when to reuse cached values or recompute attention outputs based on estimated similarity.

At every timestep, *CachedAttention* approximately evaluates the similarity between the estimated attention scores and the cached values for each input token. If the similarity exceeds a predefined threshold, we reuse the cached results, bypassing the computation. Otherwise, the attention score and output for the token are recomputed and updated in the cache. The final attention output is then aggregated from a combination of reused and recomputed outputs, leveraging the tiling properties of attention as outlined in [12].

We implement *CachedAttention* on PixArt-Alpha[9] and OmniGen[41] and conduct both accuracy and efficiency experiments using the COCO[10] and Urban1k[45] datasets. With awareness of the varying similarity of tokens across timesteps, *CachedAttention* enhances the quality of generated images compared to static cache-and-reuse methods. In the end-to-end efficiency evaluation,

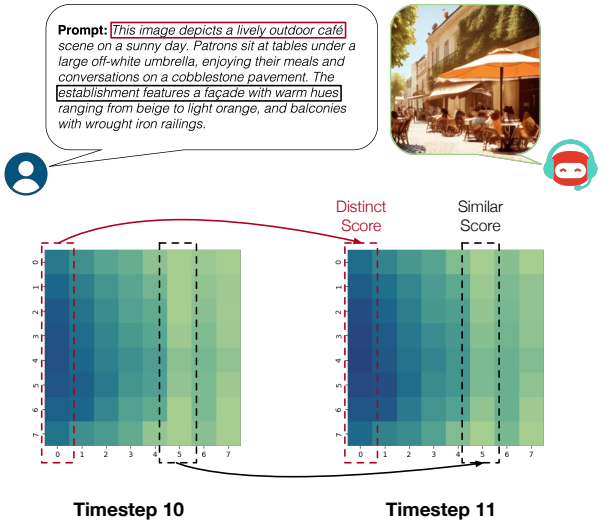


Figure 3: Attention scores of neighboring timesteps with a given prompt. Text in the red box describes the overall features, with noticeable attention score changes corresponding to the dashed box. Text in the black box describes detailed features, which are not significantly reflected at this timestep, with minimal attention score changes.

CachedAttention outperforms the baseline by up to 34%, demonstrating its effectiveness in reducing condition redundancy and accelerating conditional image generation. In summary, our contributions are as follows:

- We analyze the attention outputs in diffusion transformer models and identify condition redundancy as a key bottleneck in conditional image generation tasks, supported by a latency breakdown of the models.
- We propose *CachedAttention*, a training-free attention mechanism that adopts a dynamic cache-and-reuse policy to alleviate condition redundancy and speed up conditional image generation.
- We present comprehensive quality and efficiency evaluations of *CachedAttention*, showing up to a 34% improvement in throughput with negligible loss in image quality.

2 RELATED WORK

2.1 Image Generation

Generative Adversarial Networks (GANs) [18] have established themselves as a foundational approach in image generation research. They comprise a generator, which learns the distribution of real data to synthesize new examples, and a discriminator, which evaluates whether the input data is real or synthetic. Similarly, Variational Autoencoders (VAEs) [21] are probabilistic generative models that encode data into a latent distribution, enabling accurate reconstructions and sample generation.

In recent years, diffusion models [14, 29, 31] have emerged as a compelling alternative, offering superior stability and generation

quality compared to GANs. These models generate data by iteratively adding and removing noise, learning to reverse the noise process to produce high-quality samples while mitigating training instabilities. Diffusion models can be broadly classified into three main categories: Denoising Diffusion Probabilistic Models (DDPMs) [19, 35] utilize two Markov chains to progressively corrupt data with Gaussian noise and reverse the process by learning Markov transition kernels. Noise Conditioned Score Networks (NCSNs)[36] perturb data with multi-scale noise and estimate the score function of noisy distributions using a neural network conditioned on noise levels, enabling flexible sampling due to decoupled training and inference. Stochastic Differential Equations (SDEs) [37] extend the previous models to continuous settings, where noise perturbation and denoising follow stochastic differential equations, with probability flow ordinary differential equations (ODEs) modeling the reverse process. Early diffusion models [14, 29] were based on the U-Net architecture[32], which uses convolutional layers to model the noise schedule hierarchically.

2.2 Diffusion Transformers

Machine learning is undergoing a renaissance driven by transformers[39], which have become the dominant neural architecture across various domains over the past years. Transformers have revolutionized natural language processing[13] and numerous other fields, and image generation is no exception. To achieve greater scalability and flexibility, the transformer architecture was introduced in models like DiT [27], replacing the traditional U-Net structure. This shift has significantly improved the ability of diffusion models to capture long-range dependencies in data, further enhancing their generative capabilities.

Diffusion transformers[8, 9, 15, 41, 42] have since been applied to a wide range of tasks, including image generation. For example, PixArt-Sigma [8] demonstrates DiT's ability to produce images with high fidelity and alignment with text prompts. And OmniGen [41] introduces a unified model that not only excels at text-to-image generation but also supports diverse downstream tasks such as image editing, subject-driven generation, and visual-conditional generation. These advancements highlight the growing potential of diffusion-based models in versatile and high-quality image generation.

2.3 Acceleration Methods

Various studies have explored acceleration methods for diffusion and transformer models.

Existing methodologies aimed at accelerating both diffusion and transformer computations include a variety of optimization techniques such as distillation [17, 20, 33], quantization [16, 23, 24, 34] and pruning [7].

For transformer models, several approaches have been developed to enhance the efficiency of attention mechanisms in addition to the traditional optimization techniques mentioned above. Key-value (KV) cache stores the computed key and value pairs from previous timesteps during sequential processing, allowing the model to avoid recalculating these values at each step. This significantly speeds up inference by reducing redundant computations. However, the use of KV cache introduces significant temporal and spatial overhead in long-context scenarios. To mitigate this issue, several recent works[5, 12, 30, 38, 40, 46] have focused on compressing the KV cache to enhance attention efficiency, reduce memory usage and preserve accuracy at best effort.

FlashAttention [11, 12] accelerates self-attention using tiling and online softmax. It splits the attention matrix into smaller blocks and computes attention scores incrementally. Specifically, it decomposes the softmax of $[x_1\ x_2] \in \mathbb{R}^{2B}$ as:

$$m(x) = m([x_1\ x_2]) = \max(m(x_1), m(x_2)) \tag{1}$$

$$f(x) = \left[e^{m(x_1)-m(x)} f(x_1)\ e^{m(x_x)-m(x)} f(x_2)\right] \tag{2}$$

$$l(x) = l([x_1\ x_2]) \tag{3}$$

$$= e^{m(x_1)-m(x)} l(x_1) + e^{m(x_x)-m(x)} l(x_2) \tag{4}$$

$$\text{softmax}(x) = \frac{f(x)}{l(x)} \tag{5}$$

where

$$m(x) = \max_i x_i \tag{6}$$

$$f(x) = \left[e^{x_1-m(x)}\ \cdots\ e^{x_B-m(x)}\right] \tag{7}$$

$$l(x) = \sum_i f(x)_i \tag{8}$$

In this work, we use the tiling method introduced in FlashAttention to aggregate the output of different blocks.

Strategies to accelerate diffusion models mainly focus on efficient denoising processes, besides compression [31, 47] and distillation [17], another approach aims to improve sampling efficiency by developing training-free algorithms. Many of these methods leverage the connection between diffusion models and differential equations [37], utilizing exponential integrators to reduce sampling steps while preserving numerical accuracy.

3 METHOD

3.1 Overview

In this section, we first motivate *CachedAttention* by analyzing the condition redundancy of DiT. Then we introduce *CachedAttention* to address the problem.

3.2 Condition Redundancy

The diffusion model exhibits temporal redundancy, where the latent inputs at adjacent time steps show a high degree of similarity[22, 43]. Our key observation is that temporal redundancy also leads to condition redundancy in DiT, that the attention score between the image and the condition often shows high similarity across neighboring timesteps. This is because the attention output is computed from

$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{9}$$

where

$$Q = x_{\text{image}} W_Q \tag{10}$$

$$K = x_{\text{condition}} W_K \tag{11}$$

$$V = x_{\text{condition}} W_V \tag{12}$$

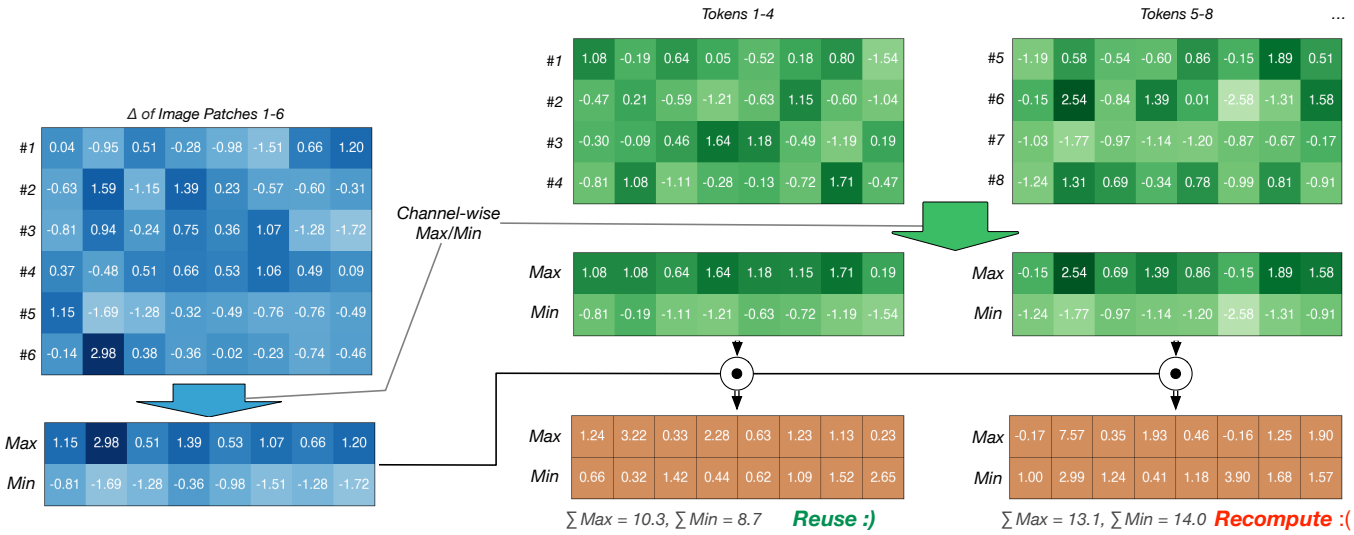


Figure 4: Estimate algorithm

that x_{image} of timestep i is similar to that of timestep $i-1$ (from temporal redundancy), and $x_{\text{condition}}$ is the same across all timesteps.

This theoretical result is further validated by our experiments. As illustrated in 3, the attention score maps at timestep 10 and timestep 11 demonstrate a high degree of similarity. Moreover, we observe that this similarity varies across different tokens. For instance, the attention scores of the fifth block of tokens are more similar between timesteps compared to that of the first tokens.

Algorithm 1 Similarity Estimation

Input: Query $\mathbf{Q}$, key $\mathbf{K}$, channel-wise max/min values of keys M_k, m_k.

Compute the difference $\Delta\mathbf{Q} = \mathbf{Q} - \mathbf{Q}_{\text{cached}}$
$M_q = \max(\Delta\mathbf{Q}, dim = -1)$
$m_q = \min(\Delta\mathbf{Q}, dim = -1)$

for $idx = 1$ **to** $blocks$ **do**
  Initialize $diff_{idx} = 0$
  for $i = 1$ **to** dim **do**
    $diff_{idx}$ += $\max(M_{q,i} * M_{k,i}, m_{q,i} * m_{k,i})$
  end for
end for
return $diff$

3.3 *CachedAttention*

To reduce condition redundancy while maintaining an accurate measure of varying similarity, we introduce *CachedAttention*, a dynamic attention mechanism that estimates the similarity of each block of tokens at every timestep and determines whether to reuse the previous output or recompute.

CachedAttention selects tokens at the granularity of blocks, which consist of several tokens, thereby minimizing the overhead associated. For each block, it employs an efficient and accurate algorithm

Algorithm 2 Blocks Aggregation

Input: Query $\mathbf{Q}$, key $\mathbf{K}$, value $\mathbf{V}$, differences $diff$, threshold θ.
Cached values: score cache $\mathbf{S}_{\text{cached}}$, output cache $\mathbf{T}_{\text{cached}}$.

Initialize $\tilde{m} = -\infty, \tilde{l} = 0$
for $i = 1$ **to** dim **do**
  if $diff_{idx} > \theta$ **then**
    $\mathbf{S}_i = \mathbf{Q}\mathbf{K}_i^T$
    $m_i = \text{rowmax}(\mathbf{S}_i)$
    $\mathbf{P}_i = \exp(\mathbf{S}_i - m_i)$
    $\mathbf{T}_i = \mathbf{T}_i\mathbf{V}_i$
    $l_i = \text{rowsum}(\mathbf{P}_i)$
  else
    Load $\mathbf{S}_i$ and $\mathbf{T}_i$ from cache
    Compute m_i, l_i as above
  end if
  $\tilde{m}_{\text{new}} = \max(\tilde{m}, m_i)$
  $\tilde{l}_{\text{new}} = e^{\tilde{m}-\tilde{m}_{\text{new}}}l + e^{m_i-\tilde{m}_{\text{new}}}l_i$
  $\mathbf{O} = (\tilde{l}_{\text{new}})^{-1}(\tilde{l} \cdot e^{\tilde{m}-\tilde{m}_{\text{new}}}\mathbf{O} + e^{m_i-\tilde{m}_{\text{new}}}\mathbf{T}_i)$
  $\tilde{m} = \tilde{m}_{\text{new}}, \tilde{l} = \tilde{l}_{\text{new}}$
end for
Update cache
return $\mathbf{O}$

to approximately evaluate the similarity between its current attention scores and the cached values, as shown in Figure 4 and Algorithm 1. The core idea is to select blocks with the maximum difference in attention score, as defined by:

$$\text{softmax}\left(\frac{Q_{\text{actual}}K^T}{\sqrt{d_k}}\right) - \text{softmax}\left(\frac{Q_{\text{cached}}K^T}{\sqrt{d_k}}\right) \tag{13}$$

By leveraging the properties of the `softmax` function, we can use the upper bound of attention weights to estimate the attention score. Motivated by Quest [38], Given the actual query Q_{actual} and

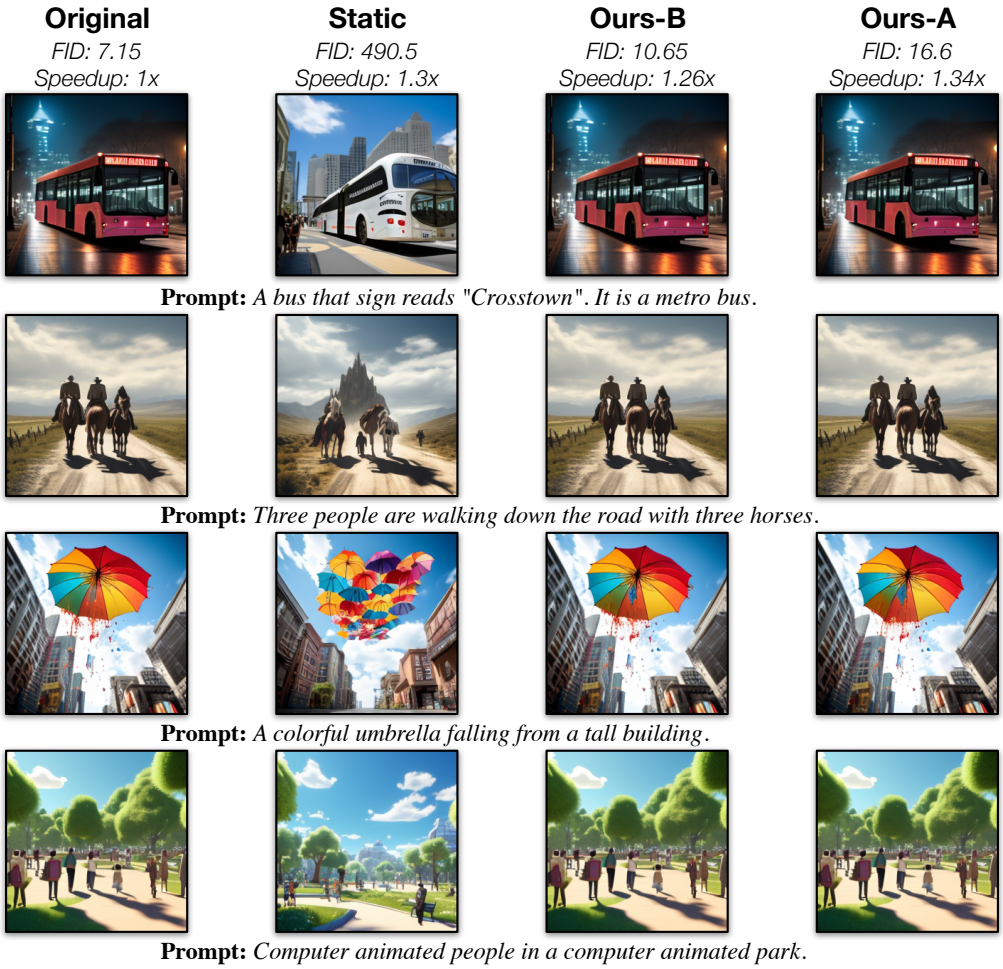


Figure 5: Quality evaluation results.

the cached value Q_{cached}, *CachedAttention* calculates the changed score of the channel i by evaluating

$$\Delta_i = \max(\max \Delta Q_i (\max K_i)^T, \min \Delta Q_i (\min K_i)^T) \tag{14}$$

where

$$\Delta Q_i = Q^i_{\text{actual}} - Q^i_{\text{cached}} \tag{15}$$

Then *CachedAttention* selects the blocks where Δ_i exceeds a given threshold, recomputes their attention scores, and reuses the cached values for the remaining blocks. Finally, the outputs of different blocks are aggregated to get the final result, as shown in Algorithm 2.

4 EXPERIMENTS

We first describe our experiment settings, including the datasets, models, and hardware configurations. Then we present our detailed evaluation results.

4.1 Settings

Datasets. We use COCO Captions 2014 [10] dataset, which contains human-generated captions for images from Microsoft Common Objects in COntext (COCO) dataset [25], to evaluate the performance of caption-to-image tasks. To simulate other tasks where the condition length is larger, We also use Urban1k, a dataset introduced in [45] that contains images with longer prompts to perform evaluations.

Models. We use two open-source conditional image generation models, PixArt-alpha [9] and OmniGen [41]. PixArt-alpha is a Transformer-based T2I diffusion model, whose image generation quality is competitive with state-of-the-art image generators. It adopts the cross-attention mechanism to align generated images with prompts. OmniGen is a Transformer-based unified image generation model. The model demonstrates competitive text-to-image generation capabilities and inherently supports a variety of downstream tasks, such as image editing and multimodal-to-image generation. Furthermore, using unified self-attention in its architecture, it can handle complex tasks end-to-end without any lengthy intermediate steps.

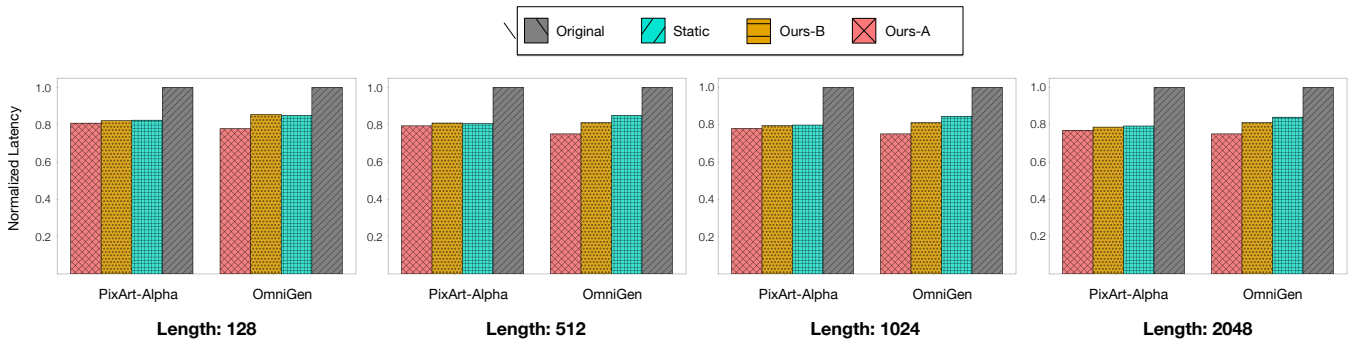


Figure 6: Efficiency evaluation results.

Baselines. We use two variants of *CachedAttention* in the experiments, where Ours-A (with a higher similarity threshold) adopts an aggressive cache-and-reuse policy, and Ours-B (with a lower similarity threshold) adopts a conservative policy. They are compared to two baselines as described in the previous discussion:

- *Original.* The original model without any optimizations. In some of the experiments, we use the generated image of this model as the standard output to perform quality evaluations.
- *Static.* The optimized implementation with static cache-and-reuse policy. It caches the cross-attention/self-attention outputs and reuses them for the next 3 timesteps.

Hardware Configurations. All the experiments in this paper are performed on an NVIDIA RTX-4090 with CUDA 12.4[1] and PyTorch 2.5.0[6].

4.2 Quality Evaluation

We first compare the quality of generated images with baselines on COCO dataset. Figure 5 shows visual results with quantitative evaluations, where Fréchet Inception Distance (FID) shows the similarity of the generated images with the standard ones (lower is better). From the evaluation results, we can observe that *CachedAttention* maintains a high consistency with the original image. *Static* achieves speedup compared to the original model, but its generated images suffer from a high proportion of information loss, with an FID value of 490.5. Our method achieves comparable speedup but has negligible loss in image quality.

4.3 Efficiency Evaluation

We then compare the efficiency of generating images with baselines in various prompt lengths. Figure 6 shows the end-to-end evaluation results on our speedup compared to all baselines. From the evaluation results, we can find that Ours-A outperforms all other variants in performance, demonstrating its efficiency in reducing condition redundancy in diffusion transformer models. The latency of Ours-B is also comparable to *Static*, but with a much high quality in generated images. What's more, we observe that as the condition length increases, the speedup of *CachedAttention* is more pronounced, which is consistent with the theoretical analysis.

5 DISCUSSION AND LIMITATION

As an acceleration method for diffusion transformers, *CachedAttention* can be further applied into tasks other than conditional image generation, e.g. video/music generation[2, 3]. Moreover, some algorithms and implementations in this work can be optimized. For example, the algorithm to select similar tokens is a heuristic-based approximation, which can be replaced by some more precise mathematical methods. We also found that due to certain implementation issues, the speedup with high condition length is not as promising as we expected. We believe the CUDA kernels implemented can be further optimized can there achieve more speedup.

6 CONCLUSION

In this paper, we present *CachedAttention*, a training-free attention mechanism for accelerating DiTs. By recognizing the varying similarity of different timesteps and tokens in the condition, *CachedAttention* dynamically caches and reuses the intermediate results in attention operators, thereby reducing redundant computations. Our extensive evaluations highlight *CachedAttention*'s ability to significantly improve overall throughput by 34%, compared with state-of-the-art solutions. These results confirm *CachedAttention*'s potential as a robust and efficient DiT optimization.

REFERENCES

[1] 2007. CUDA C++ Programming Guide. https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html.
[2] 2024. Open-Sora: Democratizing Efficient Video Production for All. https://hpcaitech.github.io/Open-Sora/.
[3] 2024. OpenAI Sora. https://openai.com/sora/.
[4] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).
[5] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245* (2023).
[6] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, et al. 2024. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*. 929–947.
[7] Thibault Castells, Hyoung-Kyu Song, Bo-Kyeong Kim, and Shinkook Choi. 2024. LD-Pruner: Efficient Pruning of Latent Diffusion Models using Task-Agnostic Insights. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 821–830.

[8] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. 2024. Pixart-\sigma: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. *arXiv preprint arXiv:2403.04692* (2024).

[9] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. 2023. Pixart-alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426* (2023).

[10] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325* (2015).

[11] Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691* (2023).

[12] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems* 35 (2022), 16344–16359.

[13] Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[14] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems* 34 (2021), 8780–8794.

[15] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. 2024. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*.

[16] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323* (2022).

[17] Zhengyang Geng, Ashwini Pokle, and J Zico Kolter. 2024. One-step diffusion distillation via deep equilibrium models. *Advances in Neural Information Processing Systems* 36 (2024).

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

[20] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351* (2019).

[21] Diederik P Kingma. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[22] Muyang Li, Tianle Cai, Jiaxin Cao, Qinsheng Zhang, Han Cai, Junjie Bai, Yangqing Jia, Kai Li, and Song Han. 2024. Distrifusion: Distributed parallel inference for high-resolution diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7183–7193.

[23] Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. 2023. Q-diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 17535–17545.

[24] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration. arXiv:2306.00978 [cs.CL] https://arxiv.org/abs/2306.00978

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 740–755.

[26] Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434* (2024).

[27] William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4195–4205.

[28] Can Qin, Shu Zhang, Ning Yu, Yihao Feng, Xinyi Yang, Yingbo Zhou, Huan Wang, Juan Carlos Niebles, Caiming Xiong, Silvio Savarese, et al. 2023. Unicontrol: A unified diffusion model for controllable visual generation in the wild. *arXiv preprint arXiv:2305.11147* (2023).

[29] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* 1, 2 (2022), 3.

[30] Luka Ribar, Ivan Chelombiev, Luke Hudlass-Galley, Charlie Blake, Carlo Luschi, and Douglas Orr. 2023. Sparq attention: Bandwidth-efficient llm inference. *arXiv preprint arXiv:2312.04985* (2023).

[31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.

[32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 234–241.

[33] V Sanh. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[34] Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2020. Q-bert: Hessian based ultra low precision quantization of bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 8815–8821.

[35] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*. PMLR, 2256–2265.

[36] Yang Song and Stefano Ermon. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems* 32 (2019).

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

[38] Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. 2024. Quest: Query-Aware Sparsity for Efficient Long-Context LLM Inference. *arXiv preprint arXiv:2406.10774* (2024).

[39] A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).

[40] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient Streaming Language Models with Attention Sinks. arXiv:2309.17453 [cs.CL] https://arxiv.org/abs/2309.17453

[41] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Shuting Wang, Tiejun Huang, and Zheng Liu. 2024. Omnigen: Unified image generation. *arXiv preprint arXiv:2409.11340* (2024).

[42] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. 2024. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528* (2024).

[43] Zhihang Yuan, Hanling Zhang, Pu Lu, Xuefei Ning, Linfeng Zhang, Tianchen Zhao, Shengen Yan, Guohao Dai, and Yu Wang. 2024. Ditfastattn: Attention compression for diffusion transformer models. *arXiv preprint arXiv:2406.08552* (2024).

[44] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems* 33 (2020), 17283–17297.

[45] Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. 2025. Long-clip: Unlocking the long-text capability of clip. In *European Conference on Computer Vision*. Springer, 310–325.

[46] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems* 36 (2023), 34661–34710.

[47] Tianchen Zhao, Tongcheng Fang, Enshu Liu, Rui Wan, Widyadewi Soedarmadji, Shiyao Li, Zinan Lin, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. 2024. ViDiT-Q: Efficient and Accurate Quantization of Diffusion Transformers for Image and Video Generation. arXiv:2406.02540 [cs.CV] https://arxiv.org/abs/2406.02540

[48] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. 2024. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039* (2024).