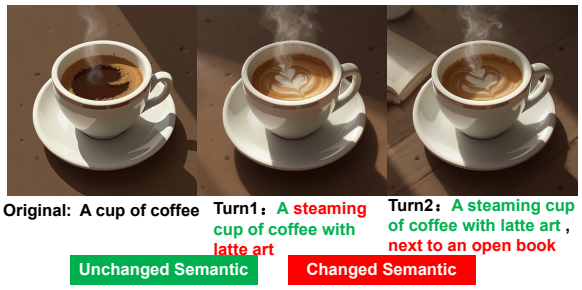
RegionCache: Semantic-Aware Region Reuse for Efficient Multi-Turn Image Generation

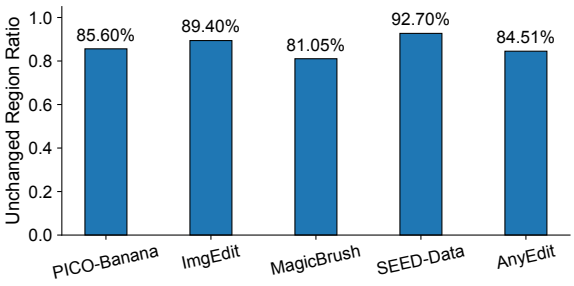
Peizheng Li, **Xin Ai**, **Hanyuan Liu**, **Qiange Wang***, **Yanfeng Zhang***
School of Computer Science and Engineering, Northeastern University, China
{lipz2, aix, liuhanyuan}@mails.neu.edu.cn, {wangqg, zhangyf}@mail.neu.edu.cn

Abstract

Real-world image generation often involves multi-turn editing, where users iteratively modify small regions while most image content remains unchanged. However, existing diffusion transformer (DiT)-based editing pipelines recompute the entire image at every turn, causing substantial redundant computation. Existing DiT acceleration methods further ignore semantic correspondence across prompts, leading to unnecessary recomputation or unsafe reuse that harms editing quality. To address this, we propose **RegionCache**, a semantic-aware reuse framework for multi-turn image editing that selectively reuses diffusion states from unchanged regions. RegionCache detects reusable regions through semantic overlap between consecutive prompts and cross-attention localization, and adopts an adaptive reuse schedule based on prompt similarity and contextual consistency. Experiments on PixArt-α demonstrate that RegionCache achieves $1.43\times$–$2.55\times$ end-to-end speedup while maintaining comparable image quality.



(a) Example of multi-turn image editing.



(b) Unchanged region ratio in image edit dataset.

Figure 1: Multi-turn image editing illustration.

1 Introduction

Image generation is a core component of modern visual content applications [Dhariwal and Nichol, 2021; Rombach *et al.*, 2022]. In practice, it is rarely a one-shot process; users typically refine results through multiple editing turns [Nichol and Dhariwal, 2021; Miyake *et al.*, 2025] until the output matches their intent. We refer to this setting as post-generation **multi-turn image editing**, where images are progressively refined by repeatedly re-running the diffusion process. Existing image editing methods adopt Diffusion Transformers (DiTs) due to their strong generation quality [Labs *et al.*, 2025; Esser *et al.*, 2024; Chen *et al.*, 2024a]. However, DiT inference is computationally expensive, and repeating it across multiple editing turns leads to substantial redundant computation overhead.

Through an analysis of real-world multi-turn image editing datasets, we identify two key observations. First, edits are typically spatially localized. In each turn, users usually modify a small, concentrated region of the image, while the remaining content is expected to stay unchanged (e.g., adjusting the coffee appearance while preserving the cup and background in Figure 1a. Across five real-world datasets [Yu *et al.*, 2025a; Ge *et al.*, 2024; Zhang *et al.*, 2023; Ye *et al.*, 2025; Qian *et al.*, 2025], we quantify cross-turn image similarity using pixel-level overlap and find that consecutive turns share 81.05%–92.70% unchanged regions on average. Second, edit instructions exhibit strong semantic persistence. Consecutive prompts largely repeat the same semantic content and differ only by a small incremental change. This semantic consistency naturally indicates which image content should be preserved across turns, while the prompt difference highlights the regions that require modification.

These observations suggest an opportunity to reduce redundant computation in multi-turn image editing by exploiting cross-turn semantic consistency. Since most of the image remains unchanged and most prompt semantics persist

*Corresponding authors.

across turns, a large portion of the computation is repeatedly spent on the same content. In particular, semantics that are repeated in consecutive prompts typically correspond to unchanged image regions. By aligning these repeated semantics with their spatial support, we can reuse intermediate diffusion states from previous turns for those regions, avoiding full-image attention recomputation while focusing updates only on regions implied by the prompt delta.

Despite the intuitive opportunity for cross-turn computation reuse, existing DiT inference acceleration methods that exploit computation reuse are primarily designed for single-pass image generation [Ma *et al.*, 2024; Zou *et al.*, 2025; Liu *et al.*, 2024]. These methods typically reduce inference cost by skipping denoising steps [Ma *et al.*, 2024; Liu *et al.*, 2024] or pruning attention [Zou *et al.*, 2025; Bolya and Hoffman, 2023] within a single editing turn, and their reuse decisions are made solely based on intra-run characteristics. As a result, they do not directly address the dominant inefficiency in multi-turn editing, where similar content repeatedly appears across successive editing turns and redundancy arises across runs. Moreover, prior works [Rombach *et al.*, 2022; Zhou *et al.*, 2025] have shown that aggressive approximation within a single inference run can degrade image quality, since high-quality editing often relies on dense attention and gradual refinement. These limitations motivate a more reliable reuse strategy tailored to multi-turn editing, which explicitly leverages cross-turn semantic consistency.

In this work, we propose **RegionCache**, an end-to-end framework for efficient multi-turn image editing that combines prompt-aligned region caching with a semantic-aware region reusing. Given a new editing prompt over a generated image, RegionCache identifies regions whose diffusion states can be safely reused by aligning shared semantics across consecutive prompts and localizing their spatial support via cross-attention maps. During inference, RegionCache reuses cached diffusion states for unchanged regions and recomputes only the regions affected by the prompt update. To balance efficiency and quality, RegionCache further determines the reuse depth based on prompt semantic similarity and editing context: semantically consistent edits permit reuse for more diffusion steps, while larger semantic changes reduce the reuse depth and fall back to reusing only early steps (e.g., *cat* $\rightarrow$ *dog*). This design avoids unnecessary recomputation while maintaining high editing fidelity in interactive, text-driven workflows.

The main contributions of this paper are summarized as follows:

- We identify cross-turn region-level semantic stability as a key source of redundant computation in multi-turn image editing, and propose a prompt-aligned region caching mechanism that localizes reusable regions via cross-attention.
- We propose a semantic-aware region reusing policy that adaptively determines the safe reuse horizon based on prompt semantic similarity and editing context, enabling aggressive reuse for semantically consistent edits and conservative reuse for larger semantic changes.
- We design **RegionCache**, an end-to-end framework for DiT-based multi-turn image editing that integrates region-level caching with adaptive step reuse, achieving significant inference speedups while preserving high editing fidelity.

2 Related Work

2.1 Diffusion Transformer Inference Acceleration Methods

Diffusion Transformers (DiTs) [Peebles and Xie, 2023; Chen *et al.*, 2024a] have become a popular backbone for high-quality image generation by modeling latent patches with global self-attention. While effective, DiT inference is computationally expensive due to the iterative denoising process and the quadratic complexity of self-attention. This has motivated a large body of work on accelerating diffusion inference.

Many acceleration methods for diffusion models exploit the observation that intermediate representations evolve smoothly across denoising steps, allowing partial reuse of computation [Zhang *et al.*, 2025a; Chen *et al.*, 2024b; Yu *et al.*, 2025b; Liu *et al.*, 2024]. For example, DeepCache [Ma *et al.*, 2024] identifies diffusion steps where feature changes are limited and reuses previously computed representations, thereby skipping redundant computation. Δ-DiT [Chen *et al.*, 2024b] introduces a Δ-Cache that stores feature deviations instead of feature maps, enabling block skipping in DiT for efficient acceleration. BlockDance [Zhang *et al.*, 2025a] accelerates diffusion inference by reusing structural features across consecutive steps and skipping the computation of shallow Transformer blocks within each denoising step.

Another line of work focuses on reducing the cost of attention computation, which dominates the runtime of Diffusion Transformers. These methods dynamically reduce attention computation by identifying or predicting which tokens contribute less to generation [Zhang *et al.*, 2025b; Yuan *et al.*, 2024; Bolya and Hoffman, 2023]. For example, TGATE [Liu *et al.*, 2024] observes that the importance of attention computation varies across diffusion stages, and reduces inference cost by selectively reusing attention outputs at stages where their contribution becomes less critical. RAS [Liu *et al.*, 2025] and ToCa [Zou *et al.*, 2025], which predict redundant tokens and bypass their attention computation.

Overall, existing DiT inference acceleration methods are primarily designed for single-pass generation. Their reuse decisions rely on intra-run characteristics, such as step-wise similarity or token importance, and do not capture redundancy that arises across successive editing turns. When applied to multi-turn editing, these methods either fail to reduce cross-turn redundancy or introduce noticeable quality degradation due to aggressive approximation.

2.2 Multi-turn Image Editing

Multi-turn image editing [Joseph *et al.*, 2024; Zhou *et al.*, 2025] supports iterative refinement, where each turn modifies an existing image to better match user intent. The process

starts from a given image—either real or previously generated—and applies a sequence of edits while preserving appropriate aspects of the image across turns. This reflects common user workflows, where images are refined over multiple turns until a satisfactory result is obtained.

Training-based methods. Training-based methods [Esser *et al.*, 2024; Wang *et al.*, 2023; Huang *et al.*, 2024] explicitly optimize diffusion models for editing by introducing additional supervision, fine-tuning, or architectural extensions (e.g., paired editing trajectories, region-aware annotations, or instruction-tuning on large synthetic datasets). Such designs often yield strong editability and robustness, but require extra data collection and training/optimization cost, making them less flexible for interactive multi-turn use.

Training-free methods. Several training-free methods [Tumanyan *et al.*, 2023; Couairon *et al.*, 2022; Cao *et al.*, 2023] address multi-turn image editing by modifying the inference behavior of pretrained diffusion models. These approaches mainly focus on improving editing controllability and visual consistency, for example by constraining attention or injecting reference information from previous generations. Prompt-to-Prompt (P2P) [Hertz *et al.*, 2022] steers cross-attention maps to localize edits while preserving unrelated content, whereas StableFlow [Avrahami *et al.*, 2025] maintains consistency by injecting reference features during inference. Region-Aware Generation (RAG) [Chen *et al.*, 2024c] further introduces region-aware constraints to guide localized generation.

However, these methods primarily operate at the output or attention guidance level and do not change the underlying inference execution. In practice, the diffusion model still performs full-image denoising and attention computation at every step, even when the edit affects only a small region. As a result, while these approaches improve editing quality and stability, they do not reduce the computational cost of multi-turn editing, and repeated full-image inference remains the dominant source of latency.

3 RegionCache

3.1 Overview

In this section, we present the design principles and key techniques of RegionCache for accelerating multi-turn image editing with Diffusion Transformers. RegionCache exploits both region-level and timestep-level reuse to reduce redundant computation across editing turns.

Figure 2 shows an overview. Specifically, RegionCache consists of two core components. First, a prompt-aligned region caching mechanism reuses intermediate diffusion states for regions whose semantics persist across consecutive prompts, while recomputing only regions implicated by the new edit. Second, a semantic-aware region reuse strategy decides how far each cached region can be reused along the denoising trajectory. It leverages prompt similarity and editing context to choose a reuse depth: semantically consistent edits allow reuse for more steps, whereas larger semantic changes restrict reuse to early steps and progressively refresh the region when needed. Together, these components enable efficient multi-turn editing while preserving generation quality.

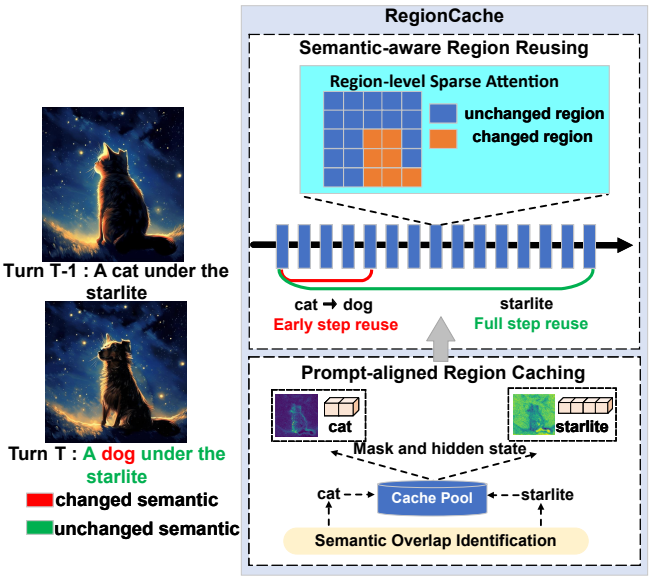


Figure 2: An overview of RegionCache.

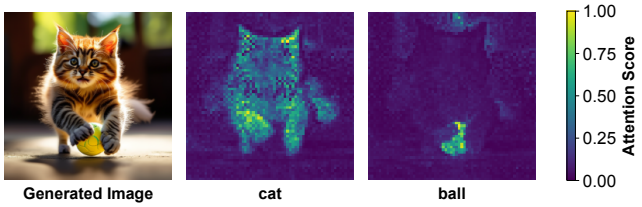


Figure 3: Token-level cross-attention visualization.

3.2 Prompt-aligned Region Caching

RegionCache localizes reusable content by exploiting semantic consistency across consecutive editing prompts. Given two prompts from successive turns, we first identify which semantics persist and which are newly introduced by comparing their tokens. Persistent semantics indicate image content that should be preserved and reused, while newly introduced or modified semantics specify where the edit should take effect. To reuse diffusion states accordingly, these semantic differences must be translated into spatial regions in the image.

Diffusion Transformers provide a natural mechanism for this translation through cross-attention. During generation, cross-attention explicitly links text tokens to the spatial locations they influence in the latent image. As illustrated in Fig. 3, tokens describing concrete objects (e.g., *cat* or *ball*) typically attend to compact and contiguous regions, with negligible attention on unrelated areas. RegionCache leverages this property to align prompt semantics with image regions, enabling region-level caching of diffusion states.

Concretely, RegionCache localizes edit regions using cross-attention maps produced during diffusion inference. Let $A_t \in \mathbb{R}^{N \times M}$ denote the cross-attention matrix at denoising step t, where N and M are the numbers of image and text tokens. For each newly introduced or modified token j, we aggregate its attention scores across heads and layers to

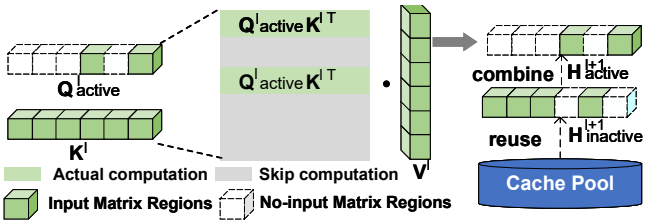


Figure 4: Region-level sparse attention in the transformer block.

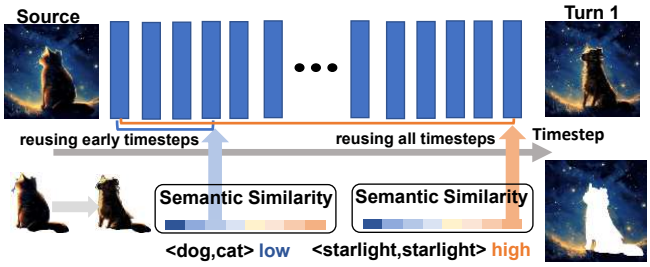


Figure 5: Adaptive timestep reuse driven by semantic similarity.

obtain a spatial attention vector $a_j \in \mathbb{R}^N$. Image tokens with salient attention responses are considered affected by the edit. Formally, the edit region is defined as

$$\mathcal{R}_{\text{edit}} = \{i \mid a_j(i) \geq \theta_j\},$$

where θ_j is an attention-based threshold derived from a_j. Regions outside $\mathcal{R}_{\text{edit}}$ correspond to persistent semantics and are treated as reusable; their cached diffusion states are preserved for subsequent editing turns.

Hidden-State-Oriented Caching. After localizing the regions affected by an edit, RegionCache avoids redundant computation by reusing intermediate hidden states of the remaining regions across editing turns. In Diffusion Transformers, hidden states are the internal representations processed at each denoising step: they are the direct inputs and outputs of self-attention and MLP layers, and fully determine the computation performed at that step. Consequently, hidden states dominate the inference cost in DiT-based editing.

Caching hidden states therefore provides a direct and effective way to reduce computation. By reusing the hidden states of semantically stable regions, RegionCache can skip repeated self-attention and MLP computation for those regions while preserving their denoising behavior. This enables reuse at the exact level where most computation occurs, without approximating or altering the model's execution.

To support multi-turn editing, RegionCache maintains a cache pool of hidden states along the diffusion trajectory. During a new editing turn, cached hidden states corresponding to unchanged regions are retrieved and reused, while hidden states of edit-affected regions are recomputed and updated. This design enables efficient cross-turn reuse with minimal impact on generation quality.

3.3 Semantic-aware Region Reusing

Region-Level Sparse Attention. To reuse cached hidden states while avoiding redundant attention computation, RegionCache introduces region-level sparse attention. The key idea is to perform self-attention only on regions affected by the current edit, while directly reusing cached hidden states for all other regions. This design preserves the standard DiT execution semantics for edited regions, while eliminating unnecessary computation on unchanged content.

As illustrated in Figure. 4, region-level sparse attention is implemented inside each Transformer block. Based on the region identification results, image tokens are partitioned into active regions that require updating and inactive regions whose hidden states are reused. For a given layer l, only the hidden states of active regions are instantiated as queries, denoted as Q^l_{active}, while the key and value matrices K^l and V^l are shared across all regions and remain unchanged.

During attention computation, only Q^l_{active} participates in the matrix multiplication with K^l, producing attention outputs for the active regions. Hidden states corresponding to inactive regions are not involved in attention at all; instead, their updated representations $H^{l+1}_{\text{inactive}}$ are directly retrieved from the cache. The outputs from attention, H^{l+1}_{active}, are then merged with the cached inactive hidden states to form the complete hidden representation H^{l+1}, which is passed to the next Transformer block following the standard DiT inference pipeline.

Semantic-aware Adaptive Reuse Scheduling. Diffusion inference progresses through a sequence of denoising steps that play different roles in image generation. Prior studies [Chen *et al.*, 2024b; Zhang *et al.*, 2025a; Agarwal *et al.*, 2024; You *et al.*, 2025] have shown that early steps primarily establish coarse structure, while later steps focus on refining object-specific details and appearance. This property implies that semantic changes do not affect all diffusion steps equally.

RegionCache leverages this observation to enable adaptive reuse along the diffusion trajectory. Even when a region undergoes a semantic edit, such as changing *cat* to *dog*, the impact of this change typically emerges at later denoising steps where fine-grained semantics are resolved. The hidden states at early steps often remain compatible and can be safely reused, whereas recomputation becomes necessary only after a certain point. In contrast, for regions whose semantics remain unchanged across editing turns, cached hidden states can be reused throughout the entire diffusion process without affecting generation quality, as illustrated in Figure 5.

Based on this insight, RegionCache adopts a semantic-aware adaptive reuse scheduling strategy that determines how long cached hidden states can be reused for each region. Specifically, for each edit-affected region, we associate the corresponding text tokens before and after the edit, denoted as τ^- and τ^+. We quantify their semantic change using cosine similarity:

$$s = \text{sim}\big(\mathbf{e}(\tau^-), \mathbf{e}(\tau^+)\big), \tag{1}$$

where $\mathbf{e}(\cdot)$ denotes the token embedding.

If the similarity score exceeds a threshold τ_s, the region is considered semantically consistent across turns, and its cached hidden states are reused throughout the entire diffusion process. Otherwise, the region is treated as undergo-

ing a semantic change, and reuse is restricted to early denoising steps only. Following prior observations that early diffusion steps mainly capture coarse structure[Chen *et al.*, 2024b; Zhang *et al.*, 2025a], we reuse cached hidden states for the first ρT denoising steps and recompute the region thereafter, where T denotes the total number of denoising steps. Based on established practice in prior work, we set the reuse ratio to a fixed value $\rho = 0.2$ in our implementation.

Formally, the reuse cutoff is defined as

$$T_{\text{reuse}} = \begin{cases} T, & s \geq \tau_s, \\ \lfloor \rho T \rfloor, & s < \tau_s. \end{cases} \tag{2}$$

This design allows RegionCache to aggressively reuse computation for semantically stable regions, while conservatively limiting reuse for larger semantic edits, achieving a favorable balance between efficiency and editing fidelity.

3.4 Implementation

Kernel Fusing. Region Sparse Attention updates keys and values only for active tokens, while reusing cached ones for the rest. A naive implementation would require extra scatter operations, introducing additional memory access and kernel overhead. To avoid this, we fuse the scatter operation into the linear projection GeMM kernel, directly writing active keys and values to their target positions. This fusion eliminates redundant kernels and ensures that sparse attention yields a practical speedup.

Pipelined Execution. Cached hidden states occupy substantial memory and are therefore stored in host (CPU) memory. To reduce the overhead of transferring cached features during inference, RegionCache overlaps cache loading with computation using a pipelined execution strategy [Harris, 2012]. While active regions are being processed in one layer, cached features for the next layer are prefetched, effectively hiding data transfer cost and improving overall efficiency.

4 Experiment

4.1 Experiment Setup

Benchmarks and Task Setting We evaluate RegionCache under a unified multi-turn image editing setting based on the PICO-Banana [Qian *et al.*, 2025] and MagicBrush [Zhang *et al.*, 2023] datasets. Each sample consists of a sequence of prompt-driven edits, where each turn incrementally refines the result from the previous turn, reflecting realistic interactive editing scenarios. Unless otherwise specified, all methods are evaluated on identical editing sequences and region specifications.

Evaluation Metrics We evaluate both efficiency and editing quality. Efficiency is measured by the total wall-clock latency required to complete an entire editing sequence, with attention latency additionally reported to analyze reductions in the dominant computational cost.

For generating quality under inference acceleration methods, we report FID [Zhang *et al.*, 2018], CLIP [Radford *et al.*, 2021], PickScore [Kirstain *et al.*, 2023], and Image Relevance (IR) to measure overall image quality and text–image semantic alignment.

Method	Latency (s)↓	CLIP$_{txt}$ ↑	CLIP$_{img}$ ↑	CLIP$_{dir}$ ↑
Stable Flow	17.46	0.3097	0.8853	0.072
P-2-P	17.19	0.2779	0.9604	0.103
RAG	32.30	0.3159	**0.9678**	**0.120**
RegionCache	**11.11**	**0.3204**	0.9233	0.101

Table 1: Quantitative comparison of multi-turn image editing methods on the MagicBrush dataset, where ↑ indicates higher is better and ↓ indicates lower is better.

For multi-turn editing performance, we adopt turn-aware CLIP-based metrics. Specifically, CLIP$_{\text{txt}}$ measures prompt alignment at each editing turn, CLIP$_{\text{img}}$ evaluates cross-turn visual consistency between consecutive outputs, and CLIP$_{\text{dir}}$ assesses whether edits follow the intended semantic direction.

Single-turn acceleration baselines. We compare RegionCache with representative inference acceleration and caching methods originally designed for single-turn generation, including ToCa [Zou *et al.*, 2025], DeepCache [Ma *et al.*, 2024], TGATE [Liu *et al.*, 2024], and step-reduced PixArt-LCM [Chen *et al.*, 2023], all implemented on the same PixArt backbone for fair comparison. To evaluate these methods in a multi-turn setting, we adopt a unified editing protocol in which, at each turn, edits are applied to specified regions while unedited regions are preserved by constraining these methods to reuse and maintain the corresponding latent representations.

Multi-turn image editing baselines. We further compare RegionCache with representative training-free multi-turn image editing frameworks, including StableFlow [Avrahami *et al.*, 2025], RAG (Region-Aware Generation) [Chen *et al.*, 2024c], and Prompt-to-Prompt (P-2-P) [Hertz *et al.*, 2022], under the same multi-turn editing protocol. This comparison focuses on evaluating efficiency and editing quality under realistic interactive editing scenarios.

All methods are evaluated using the pretrained PixArt-α Diffusion Transformer under identical inference settings to ensure fair comparison. For caching-based acceleration baselines originally designed for single-turn generation, we adapt their strategies to the multi-turn setting by applying caching independently at each editing turn. All experiments are conducted on a single NVIDIA RTX L20 GPU, and results are reported over 30,000 randomly sampled multi-turn editing sequences unless otherwise specified.

4.2 Comparison with Multi-turn Image Editing Frameworks

As shown in Table 1, RegionCache achieves the lowest overall latency, completing a full multi-turn editing sequence in 11.11 s. This corresponds to a $1.6\times$ speedup over Stable Flow (17.46 s) and P-2-P (17.19 s), and a $2.9\times$ speedup over RAG (32.30 s). For Stable Flow and P-2-P, the additional runtime mainly stems from recomputing full-image self-attention at every editing turn, which introduces substantial redundant computation when large regions remain unchanged. On the other hand, RAG incurs higher latency due to the overhead introduced by explicit region-aware control.

RegionCache also maintains competitive editing quality 6. It achieves the highest CLIP$_{\text{txt}}$ score (0.3204), indicating su-

Figure 6: Case study of multi-turn image editing comparing RegionCache with existing multi-turn editing frameworks.

perior alignment with textual prompts across editing turns. This improvement stems from RegionCache's region-level reuse mechanism, which preserves semantically consistent representations for unchanged regions, thereby reducing unintended semantic drift during successive edits. While its CLIP_{img} score is slightly lower than that of P-2-P and RAG, the difference remains moderate, suggesting that RegionCache preserves cross-turn visual consistency while allowing necessary local modifications to accommodate updated prompts. In addition, RegionCache attains a comparable CLIP_{dir} score, reflecting stable and coherent editing directions, as semantic changes are explicitly constrained to edited regions.

Overall, these results indicate that RegionCache strikes a favorable balance between efficiency and editing quality.

4.3 Comparison with Single-turn Acceleration Methods

We also compare RegionCache with existing diffusion acceleration methods under a unified multi-turn setting. As shown in Table 2, RegionCache achieves a favorable balance between efficiency and image quality, outperforming the majority of compared acceleration methods in both inference speed and editing fidelity. RegionCache reduces the total inference latency to 2.53s, achieving a $1.57\times$ speedup over the full PixArt-α baseline (3.98s), while lowering the cumulative self-attention computation time across all denoising steps to 0.55 s, corresponding to a $2.89\times$ reduction. Specifically, RegionCache is faster than ToCa by 0.02 s in total inference time and reduces the cumulative self-attention cost by 0.25 s (0.55 s vs. 0.80 s), as region-level reuse enables a higher effective degree of attention sparsification across editing turns. Compared with TGATE, RegionCache further achieves a 0.15 s reduction in total inference time (2.53 s vs. 2.68 s) and lowers the attention cost by 0.34 s (0.55 s vs. 0.89 s), despite TGATE incorporating partial reuse of attention computation results, as TGATE still retains full self-attention at selected denoising steps. DeepCache attains a higher total inference speedup of $1.81\times$ by aggressively skipping diffusion steps, and therefore surpasses RegionCache in raw speedup. However, this aggressive step skipping comes at a clear cost in editing quality: DeepCache yields substantially worse FID and IR than RegionCache.

Method	Inference Time		Image Quality			
	Attn. (s)↓	Total (s)↓	FID↓	CLIP↑	IR↑	PICK↑
PixArt-α (Full compute)	1.58	3.98	**30.30**	**0.311**	0.747	**22.41**
ToCa (N=3, R=60%)	0.80	2.55	36.65	0.303	0.172	21.02
DeepCache (N=3)	0.73	**2.20**	39.65	0.307	0.186	21.14
TGATE (m=10)	0.89	2.68	31.27	0.211	0.678	21.99
PixArt-LCM (8)	0.44	1.03	33.44	0.307	0.485	21.98
RegionCache	**0.55**	2.53	**30.98**	**0.311**	**0.776**	**22.36**

Table 2: Comparison with existing multi-turn image editing methods on the PICO-Banana dataset. ↑ indicates higher is better, while ↓ indicates lower is better.

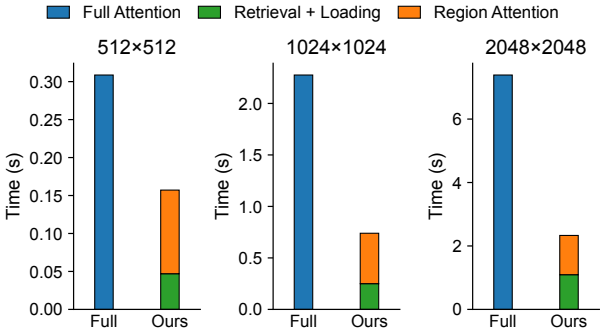


Figure 7: Resolution-wise inference overhead analysis.

RegionCache achieves the best overall image quality because it reuses a more stable caching target in multi-turn editing: hidden states from regions whose semantics persist across turns. As shown in Table 2, RegionCache attains the lowest FID score and the highest CLIP, IR, and PICK metrics, indicating superior semantic alignment and fine-grained editing fidelity. This advantage is further illustrated in Figure 8, where RegionCache produces more coherent local edits and preserves better structural consistency across multiple editing turns.

4.4 Runtime Breakdown Analysis

In a single inference pass, RegionCache introduces two additional sources of overhead: the cost of retrieving reusable cached regions and the cost of loading the corresponding cached hidden states from host memory into device memory.

Figure 7 compares these overheads with the reduction in attention computation across different image resolutions. As resolution increases, the cost of full attention grows rapidly, from around 0.3 s at 512×512 to over 2 s at 1024×1024, and exceeds 6 s at 2048×2048. In contrast, the combined retrieval and loading overhead remains small, on the order of 0.05 s at 512×512 and below 0.3 s even at 2048×2048. Across all resolutions, RegionCache reduces the attention computation by a large margin, with region-level attention accounting for only a fraction of the full attention cost. As a result, the reduction in attention computation consistently outweighs the introduced overhead, leading to a net decrease in inference time. These results indicate that RegionCache effectively targets the dominant computation while introducing only minor

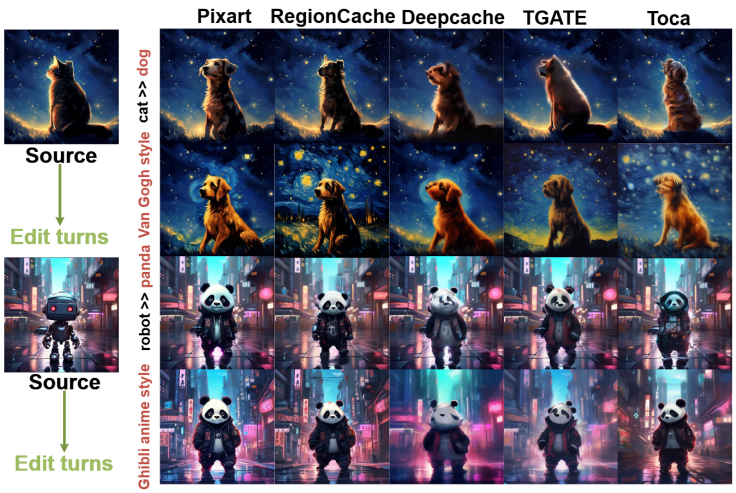


Figure 8: Case study of multi-turn image editing comparing RegionCache with existing diffusion acceleration methods.

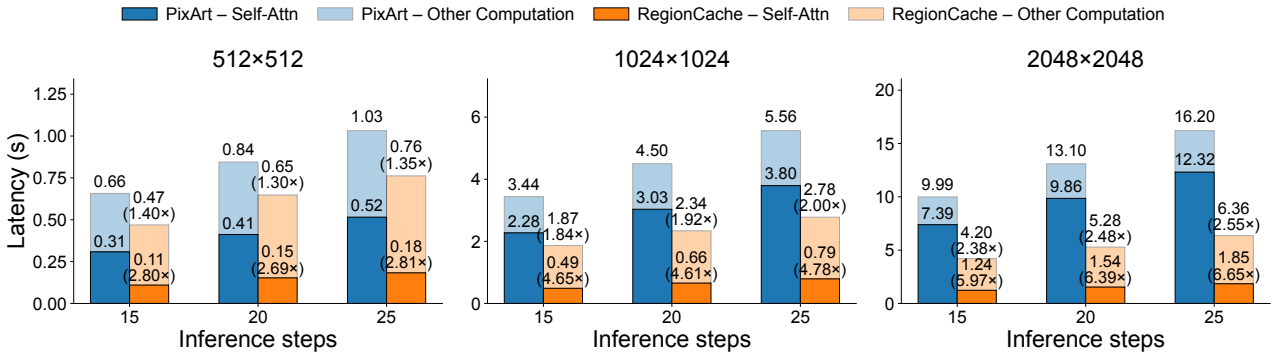


Figure 9: Latency breakdown of self-attention and other computation across different inference steps and resolutions.

additional cost.

4.5 Sensitive Study

Performance with varying steps For a fixed image resolution, increasing the number of diffusion steps primarily scales the absolute runtime of all computation components in a nearly linear manner. As shown in Figure 9, at 512×512 resolution, the self-attention latency of PixArt increases from 0.31 s at 15 steps to 0.52 s at 25 steps, while RegionCache increases from 0.11 s to 0.18 s over the same range. Despite this increase in absolute cost, the relative reduction in self-attention introduced by RegionCache remains stable at approximately $2.8\times$–$2.9\times$ across 15, 20, and 25 steps. A similar trend is observed at higher resolutions. This shows that varying the number of diffusion steps has a limited impact on the relative speedup achieved by RegionCache.

Performance with varying image resolution Image resolution has a pronounced impact on the effectiveness of RegionCache. In Figure 9, as resolution increases, self-attention rapidly becomes the dominant component of inference cost. At 15 diffusion steps, the self-attention latency of PixArt increases from 0.31 s at 512×512 to 7.39 s at 2048×2048, accounting for the majority of total runtime. Although RegionCache applies a similar relative reduction to self-attention across resolutions, the absolute benefit grows substantially with image size. Specifically, the attention speedup increases from approximately $2.7\times$ at 512×512 to $4.7\times$ at 1024×1024, and exceeds $6\times$ at 2048×2048.

This amplification stems from the increased dominance of self-attention at larger sequence lengths. As self-attention occupies a larger portion of the total computation, the same relative sparsification ratio yields a greater reduction in end-to-end latency.

5 Conclusion

We presented **RegionCache**, a region-level reuse framework for efficient multi-turn image editing with Diffusion Transformers. RegionCache identifies which parts of an image can be reused across editing turns by aligning repeated prompt semantics to image regions via cross-attention, and caches hidden states at the level where attention computation dominates. During editing, it selectively reuses cached hidden states for semantically stable regions and recomputes only the regions affected by new edits, with an adaptive reuse schedule that controls how long reuse remains valid along the diffusion trajectory. Extensive experiments demonstrate that RegionCache significantly reduces redundant attention computation and achieves consistent end-to-end speedups while preserving editing quality.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62461146205, U2241212, 625B2041), and the Distinguished Youth Foundation of Liaoning Province (2024021148-JH3/501).

References

[Agarwal *et al.*, 2024] Shubham Agarwal, Subrata Mitra, Sarthak Chakraborty, Srikrishna Karanam, Koyel Mukherjee, and Shiv Kumar Saini. Approximate caching for efficiently serving {Text-to-Image} diffusion models. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, pages 1173–1189, 2024.

[Avrahami *et al.*, 2025] Omri Avrahami, Or Patashnik, Ohad Fried, Egor Nemchinov, Kfir Aberman, Dani Lischinski, and Daniel Cohen-Or. Stable flow: Vital layers for training-free image editing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7877–7888, 2025.

[Bolya and Hoffman, 2023] 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 4599–4603, 2023.

[Cao *et al.*, 2023] 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. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22560–22570, 2023.

[Chen *et al.*, 2023] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *Proceedings of the Twelfth International Conference on Learning Representations*, 2023.

[Chen *et al.*, 2024a] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-σ: Weak-to-strong training of diffusion transformers for 4k text-to-image generation. In *European Conference on Computer Vision*, pages 74–91, 2024.

[Chen *et al.*, 2024b] Pengtao Chen, Mingzhu Shen, Peng Ye, Jianjian Cao, Chongjun Tu, Christos-Savvas Bouganis, Yiren Zhao, and Tao Chen. Δ-DiT: A Training-Free Acceleration Method Tailored for Diffusion Transformers. *arXiv preprint arXiv:2406.01125*, 2024.

[Chen *et al.*, 2024c] Zhennan Chen, Yajie Li, Haofan Wang, Zhibo Chen, Zhengkai Jiang, Jun Li, Qian Wang, Jian Yang, and Ying Tai. Region-aware text-to-image generation via hard binding and soft refinement. *arXiv preprint arXiv:2411.06558*, 2024.

[Couairon *et al.*, 2022] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. In *The Eleventh International Conference on Learning Representations*, 2022.

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

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

[Ge *et al.*, 2024] Yuying Ge, Sijie Zhao, Chen Li, Yixiao Ge, and Ying Shan. Seed-data-edit technical report: A hybrid dataset for instructional image editing. *arXiv preprint arXiv:2405.04007*, 2024.

[Harris, 2012] Mark Harris. How to overlap data transfers in cuda c/c++. *URL http://devblogs. nvidia. com/parallelforall/how-overlap-data-transfers-cuda-cc*, 2012.

[Hertz *et al.*, 2022] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. Prompt-to-prompt image editing with cross-attention control. In *The Eleventh International Conference on Learning Representations*, 2022.

[Huang *et al.*, 2024] Nisha Huang, Yuxin Zhang, Fan Tang, Chongyang Ma, Haibin Huang, Weiming Dong, and Changsheng Xu. Diffstyler: Controllable dual diffusion for text-driven image stylization. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[Joseph *et al.*, 2024] KJ Joseph, Prateksha Udhayanan, Tripti Shukla, Aishwarya Agarwal, Srikrishna Karanam, Koustava Goswami, and Balaji Vasan Srinivasan. Iterative multi-granular image editing using diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 8107–8116, 2024.

[Kirstain *et al.*, 2023] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

[Labs *et al.*, 2025] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025.

[Liu *et al.*, 2024] Haozhe Liu, Wentian Zhang, Jinheng Xie, Francesco Faccio, Mengmeng Xu, Tao Xiang, Mike Zheng Shou, Juan-Manuel Perez-Rua, and Jürgen Schmidhuber. Faster diffusion via temporal attention decomposition. *arXiv preprint arXiv:2404.02747*, 2024.

[Liu *et al.*, 2025] Ziming Liu, Yifan Yang, Chengruidong Zhang, Yiqi Zhang, Lili Qiu, Yang You, and Yuqing Yang. Region-adaptive sampling for diffusion transformers. *arXiv preprint arXiv:2502.10389*, 2025.

[Ma *et al.*, 2024] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15762–15772, 2024.

[Miyake *et al.*, 2025] Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2063–2072. IEEE, 2025.

[Nichol and Dhariwal, 2021] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

[Peebles and Xie, 2023] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[Qian *et al.*, 2025] Yusu Qian, Eli Bocek-Rivele, Liangchen Song, Jialing Tong, Yinfei Yang, Jiasen Lu, Wenze Hu, and Zhe Gan. Pico-banana-400k: A large-scale dataset for text-guided image editing. *arXiv preprint arXiv:2510.19808*, 2025.

[Radford *et al.*, 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[Rombach *et al.*, 2022] 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.

[Tumanyan *et al.*, 2023] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1921–1930, 2023.

[Wang *et al.*, 2023] Zhizhong Wang, Lei Zhao, and Wei Xing. Stylediffusion: Controllable disentangled style transfer via diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7677–7689, 2023.

[Ye *et al.*, 2025] Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. Imgedit: A unified image editing dataset and benchmark. *arXiv preprint arXiv:2505.20275*, 2025.

[You *et al.*, 2025] Haoran You, Connelly Barnes, Yuqian Zhou, Yan Kang, Zhenbang Du, Wei Zhou, Lingzhi Zhang, Yotam Nitzan, Xiaoyang Liu, Zhe Lin, et al. Layer- and timestep-adaptive differentiable token compression ratios for efficient diffusion transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18072–18082, 2025.

[Yu *et al.*, 2025a] Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26125–26135, 2025.

[Yu *et al.*, 2025b] Zichao Yu, Zhen Zou, Guojiang Shao, Chenwei Zhang, Shengze Xu, Jie Huang, Feng Zhao, Xiaodong Cun, and Wenyi Zhang. Ab-cache: Training-free acceleration of diffusion models via adams-bashforth cached feature reuse. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 10408–10417, 2025.

[Yuan *et al.*, 2024] Zhihang Yuan, Hanling Zhang, Lu Pu, Xuefei Ning, Linfeng Zhang, Tianchen Zhao, Shengen Yan, Guohao Dai, and Yu Wang. Ditfastattn: Attention compression for diffusion transformer models. *Advances in Neural Information Processing Systems*, 37:1196–1219, 2024.

[Zhang *et al.*, 2018] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[Zhang *et al.*, 2023] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023.

[Zhang *et al.*, 2025a] Hui Zhang, Tingwei Gao, Jie Shao, and Zuxuan Wu. Blockdance: Reuse structurally similar spatio-temporal features to accelerate diffusion transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12891–12900, 2025.

[Zhang *et al.*, 2025b] Jintao Zhang, Haoxu Wang, Kai Jiang, Shuo Yang, Kaiwen Zheng, Haocheng Xi, Ziteng Wang, Hongzhou Zhu, Min Zhao, Ion Stoica, et al. Sla: Beyond sparsity in diffusion transformers via fine-tunable sparse-linear attention. *arXiv preprint arXiv:2509.24006*, 2025.

[Zhou *et al.*, 2025] Zijun Zhou, Yingying Deng, Xiangyu He, Weiming Dong, and Fan Tang. Multi-turn consistent image editing. *arXiv preprint arXiv:2505.04320*, 2025.

[Zou *et al.*, 2025] Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. In *The Thirteenth International Conference on Learning Representations*, 2025.