

EEdit : Rethinking the Spatial and Temporal Redundancy for Efficient Image Editing

Supplementary Material

622

7. Analysis on Hidden States in MM-DiT

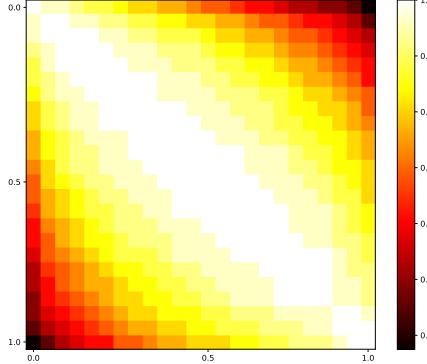


Figure 9. Cross-Attention hidden states similarity

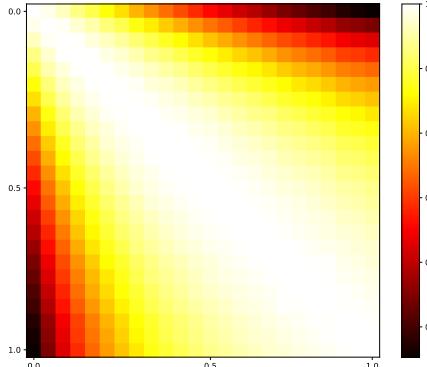


Figure 10. Self-Attention hidden states similarity

623
624
625
626
627
628
629

As shown in the Figure. 9 and Figure. 10, we visualize the cosine similarity of hidden states across different timesteps in both Cross-Attention and Self-Attention. It can be observed that Cross-Attention exhibits higher similarity, indicating greater redundancy in this module. Consequently, in our approach, Cross-Attention is either fully computed or entirely skipped to optimize efficiency.

630

8. Algorithm for Image Editing with SLoC

631

8.1. Cache Frequency Control

632

Cache Frequency Control can be formulated as

$$S_{ij}^{\tau+1} \leftarrow \begin{cases} S_{ij}^{\tau} + \gamma f_{ij}^{\tau} & f_{ij}^{\tau+1} \leftarrow f_{ij}^{\tau} + 1 \\ & S_{ij}^{\tau} \leq TopR\%(\mathcal{S}^{\tau}[1 \dots L_{WH}]) \\ S_{ij}^{\tau} & f_{ij}^{\tau+1} \leftarrow 0 \\ & S_{ij}^{\tau} > TopR\%(\mathcal{S}^{\tau}[1 \dots L_{WH}]) \end{cases} \quad 633$$

where γ is a scaling factor that controls the impact of reuse frequency on the score map \mathcal{S} . τ indicates current time step and $\tau + 1$ indicates next time step. ij indicated index in score map \mathcal{S} and frequency map \mathcal{M}_{freq} . We have $\mathcal{M}_{freq} = \{f_{ij}, i \in [1 \dots L_W], j \in [1 \dots L_H]\}$

8.2. Vanilla SLoC w/o ISS and TIP

634
635
636
637
638

Algorithm 2 Image Editing with SLoC

Require: Input image \mathbf{I}_s , Mask for editing region \mathbf{M}_s , Prompt for editing \mathbf{P}_m , Randomly initialized map \mathcal{R} , Bonus for edited region \mathbf{S}_E and Cache dict $\mathcal{C}_{l,m}[:]$.

Ensure: The edited result \mathbf{I}^*

```

1:  $\mathcal{M}_{freq} \leftarrow zero[\mathcal{F}_i, t, l]$ 
2:  $\mathbf{Z}_0 \leftarrow \text{cat}(\text{VQ-Encoder}(\mathbf{I}_s), \text{Txt-Encoder}(\mathbf{P}_m))$ 
3: // Image Latent Inversion
4: for  $t = 1, \dots, T$  do
5:    $\mathbf{Z}_t \leftarrow \text{RF-inversion}(\mathbf{Z}_{t-1}, t-1, \phi)$ 
6: end for
7:  $\mathbf{Z}_T^* \leftarrow \mathbf{Z}_T$ 
8: // Image Editing Steps with Caching
9: for  $t = T, T-1, \dots, L+1$  do
10:   for  $\mathcal{F}_i \leftarrow \mathbf{SA}_l, \mathbf{CA}_l, \mathbf{MLP}_l, l \in [1, \dots, \mathcal{L}]$  do
11:      $\mathcal{S}_l \leftarrow (\mathcal{R} \odot \mathbf{S}_E) \oplus \mathcal{M}_{freq}$ 
12:      $\mathcal{I}_{i,l,t} \leftarrow \text{Sel}_{topR\%}(\mathcal{S}_l)$ 
13:      $\mathbf{Z}_{t+1}^* \leftarrow \text{scatter}(\mathcal{F}_i(\mathbf{Z}_l^*, \mathcal{I}_{i,l,t}), \mathcal{C}_{t+1}[l, \mathcal{F}_i])$ 
14:      $\text{Update}(\mathcal{C}_{t+1}[l, \mathcal{F}_i], \mathcal{M}_{freq})$ 
15:   end for
16:    $\mathbf{Z}_{t-1}^* \leftarrow \mathbf{Z}_{t-1}^* \odot \mathbf{M}_s + \mathbf{Z}_t \odot (1 - \mathbf{M}_s)$ 
17: end for
18:  $\mathbf{I}^* \leftarrow \text{VQ-Decoder}(\mathbf{x}_0)$ 
19: return  $\mathbf{I}^*$ 

```

9. Discussion on TIP

640
641
642

We adopt Token Index Preprocessing. This design offers an additional advantage by reducing the number of function

Table 4. **Ablation study for cache.** Comparisons of different cache methods in terms of FG fidelity, and computational efficiency.

Method	FG preservation				Efficiency				Speed Up		
	FID \downarrow	PSNR \uparrow	MSE $_{10-3}^{\downarrow}$	SSIM $_{10-1}^{\uparrow}$	Steps	Inv.(s) \downarrow	Fwd.(s) \downarrow	Inference (s) \downarrow	FLOPs(T) \downarrow	Latency \uparrow	FLOPs \uparrow
No Cache	-	-	-	-	28	5.44	5.67	11.30	932.95	1 \times	1 \times
50% Step	90.60	24.46	6.66	8.77	14	2.63	2.87	5.69	456.55	2.04 \times	1.99 \times
FORA [35]	39.96	31.62	1.74	9.47	28	2.42	2.45	5.05	318.09	2.24 \times	2.93 \times
ToCa [50]	84.77	26.16	5.76	8.88	28	3.52	3.52	7.29	332.93	1.55 \times	2.80 \times
DuCa [51]	84.85	26.16	5.76	8.88	28	3.26	3.26	6.87	313.00	1.67 \times	2.98 \times
SLoC	39.50	31.75	1.72	9.48	28	2.86	2.91	5.96	384.03	1.90 \times	2.43 \times
SLoC_{+TIP+ISS}	39.21	31.75	1.71	9.48	28	1.92	2.49	4.60	264.50	2.46 \times	3.53 \times

643 calls within the cache module. Specifically, with a prepro-
644 cessing overhead of no more than **150ms**, we achieve a re-
645 duction of over **1000ms** in cache-induced inference latency.
646 Since the token selection in our algorithm is independent of
647 the internal properties of individual tokens or their mutual
648 interactions, this decoupling is logically equivalent in the
649 temporal sequence. Consequently, it enables further loss-
650 less acceleration on top of SLoC.

9.1. Proof of TIP Equivalence with Original Operations

653 We maintain a cache of intermediate features for a set of to-
654 kens in SLoC. At each iteration step, each token is assigned
655 a score based on (i) a random or seed-based component and
656 (ii) a function of its selection frequency (the \mathcal{M}_{freq}). The
657 top $R\%$ of tokens are selected for updating the cache. It
658 follows algorithm 2.

659 We prove that this preprocessing-based approach is
660 mathematically equivalent to performing scoring, sorting,
661 and token selection *online* at each iteration.

9.2. Notation and Problem Setup

- N : Total number of tokens, indexed as $\{1, 2, \dots, N\}$.
- T : Total number of diffusion iterative steps.
- $s_i^{(t)}$: Score of token i at step t , given by

$$s_i^{(t)} = f(r_i^{(t)}) + \mathcal{M}_{freq,i}^{(t)}$$

667 where:

- $r_i^{(t)}$: Random (or seed-based) component.
- $\mathcal{M}_{freq,i}^{(t)}$: Frequency of times token i has been selected
668 before step t .
- $f(\cdot)$: A deterministic function adjusting scores, region
669 score bonus adopted in SLoC here.
- After computing $\{s_i^{(t)}\}_{i=1}^N$, the top $R\%$ tokens are se-
670 lected for cache updates.
- For simplicity in our proof, we have omitted the layer
671 index and module type (Cross-Attention, Self-Attention,
672 MLP).

9.3. Original (Online) Algorithm Description

673 The online method iterates as follows 2:

1. For $t = 1$ to T :
 - Compute $s_i^{(t)}$ for each token i .
 - Sort tokens by $s_i^{(t)}$ and select the top $R\%$.
 - Update the cache for these selected tokens.
 - Increment $\mathcal{M}_{freq,i}^{(t+1)}$ for each selected token i .

685 Here, $r_i^{(t)}$ is reproducible when using a fixed seed.

9.4. Proposed Optimization (TIP)

687 The optimized approach precomputes the cache update and
688 selection process 1:

1. Generate and iterate all $r_i^{(t)}$ for $i = 1, \dots, N$ and $t = 689$
 $1, \dots, T$.
2. Simulate the selection process offline:
 - Initialize $\mathcal{M}_{freq,i}^{(1)} = 0$ for all i .
 - For each $t = 1$ to T :
 - Compute $s_i^{(t)} = f(r_i^{(t)}) + \mathcal{M}_{freq,i}^{(t)}$.
 - Sort and select the top $R\%$, recording indices as
 $\mathcal{I}_{top}^{(t)}$.
 - Update $\mathcal{M}_{freq,i}^{(t+1)}$ for selected tokens.
 3. Store $\{\mathcal{I}_{top}^{(t)}\}_{t=1}^T$ for later use.

699 At inference, we read precomputed $\mathcal{I}_{top}^{(t)}$ instead of recom-
700 puting scores.

9.5. Proof of Equivalence

702 We prove that both methods select identical tokens at each
703 step.

Step 1 Equivalence. At $t = 1$, we have $\mathcal{M}_{freq,i}^{(1)} = 0$, so

$$s_i^{(1)} = f(r_i^{(1)}) + 0.$$

706 Since $r_i^{(1)}$ is identical in both methods (fixed seed), sort-
707 ing $s_i^{(1)}$ gives the same top $R\%$ tokens, ensuring identical
708 updates and increments for $\mathcal{M}_{freq,i}^{(2)}$.

Inductive Hypothesis. Assume for steps $k < t$ that

$$\mathcal{I}_{top}^{(k)}(\text{offline}) = \mathcal{I}_{top}^{(k)}(\text{online}).$$

711 Thus, $\mathcal{M}_{freq,i}^{(t)}$ is identical in both methods.

712 **Step t Equivalence.** At step t ,

713
$$s_i^{(t)} = f(r_i^{(t)}) + \mathcal{M}_{freq,i}^{(t)}$$

714 Since $r_i^{(t)}$ and $\mathcal{M}_{freq,i}^{(t)}$ are identical (by induction), we get

715
$$s_i^{(t)}(\text{offline}) = s_i^{(t)}(\text{online}),$$

716 ensuring that sorting and selecting the top $R\%$ gives identi-
717 cal indices sets:

718
$$\mathcal{I}_{\text{top}}^{(t)}(\text{offline}) = \mathcal{I}_{\text{top}}^{(t)}(\text{online}).$$

719 **Conclusion by Induction.** By induction, token selection
720 and cache updates remain identical for all $t = 1, \dots, T$.
721 Thus, preprocessing achieves the same outcome as the on-
722 line approach.

723 10. Implementation Details

724 The experiments were conducted on a machine with the fol-
725 lowing hardware and software specifications:

726 10.1. Hardware Specifications

- 727 **Architecture:** x86_64
- 728 **CPU Op-Modes:** 32-bit, 64-bit
- 729 **Address Sizes:** 52 bits physical, 48 bits virtual
- 730 **Byte Order:** Little Endian
- 731 **Total CPU(s):** 128
- 732 **On-line CPU(s) List:** 0-127
- 733 **Vendor ID:** AuthenticAMD
- 734 **Model Name:** AMD EPYC 9K84 96-Core Processor
- 735 **CPU Family:** 25

736 10.2. Software Specifications

- 737 **Operating System:** Ubuntu 22.04.3 LTS
- 738 **Python:** 3.12.3
- 739 **huggingface-hub:** 0.26.2
- 740 **numpy:** 1.26.4
- 741 **torch:** 2.5.1
- 742 **torchmetrics:** 1.6.1
- 743 **transformers:** 4.46.1

744 11. Metrics

745 Our experiments employ a selection of the most widely used
746 image quality, instruction adherence, and efficiency metrics.

747 **Frechet Inception Distance (FID)** and **Learned Per-**
748 **ceptual Image Patch Similarity (LPIPS)** are feature-
749 based similarity metrics computed using pretrained neural
750 networks. Lower values indicate higher similarity. We use
751 InceptionV3 for FID and AlexNet for LPIPS mea-
752 surements. **Peak Signal-to-Noise Ratio (PSNR)** and **Mean
753 Squared Error (MSE)** are pixel-space similarity metrics.



Figure 11. A qualitative example. Token index preprocessing shows loss-less acceleration for editing quality.

A higher PSNR and a lower MSE indicate greater image similarity. **CLIPScore** measures how well an image generation or editing result follows a given prompt using a pretrained CLIP model. A higher score indicates stronger adherence to the prompt. In our experiments, we use the `clip-vit-base-patch16` model. **FLOPs** quantify the computational cost associated with model inference. A higher value indicates greater computational overhead.

762 12. More Experiments

Table 5. **Performance comparison.** An ablation study is conducted on imbalanced inversion and denoising for background preservation, foreground fidelity and inference time.

Inversion	Denoising	BG Preservation		FG Fidelity		Inference ↑ Time (s)
		LPIPS $\downarrow \times 10^{-2}$	LPIPS $\downarrow \times 10^{-3}$	PSNR \uparrow	FID \downarrow	
Full Step	Full Step	1.98	-	-	-	13.27
2-step skip		31.38	1.98	31.93	3.35	10.16
3-step skip	Full Step	31.38	1.98	25.79	3.23	9.31
4-step skip		31.38	1.98	26.50	3.31	8.76
	2-step skip	1.97	50.40	31.93	28.51	11.79
Full Step	3-step skip	1.97	121.44	25.79	64.10	9.28
	4-step skip	1.96	102.67	26.50	56.93	8.54

We compared the edited results in terms of image similarity and efficiency. The Table. 4 demonstrate that our approach, when incorporating all optimizations (TIP + ISS), achieves the best performance in both fidelity and efficiency. SLoC achieves a **2.46 \times** significant improvement in inference latency and a **3.53 \times** acceleration in computational efficiency compared to the original unaccelerated version. Additionally, our method demonstrates either improved fidelity or remains at a state-of-the-art level across various editing tasks.

773 13. Gallery

We present additional editing results, including prompt-guided, drag-guided, and reference-guided editing. Furthermore, we adapted the community-developed Redux model for img2img tasks, enabling the generation of impressive image variations.

Prompt-guided



A red apple and a bird sitting on it A golden pagoda in the rain Photo of a horse and a cat standing on rocks near the ocean A closed eyes cat sitting on wooden floor



A beautiful woman with hat on head A cute dog holding a pink heart A woman with monster around her face A painting of a car in the snow with mountains in the background

Drag-guided



A photo of goats A photo of a dog and a cat Majestic lion basking in the sunlight. Marble bust of a young Roman noble



an oil painting of a female A photo of a man holding a crocodile Pastoral scene with horses and dogs in a countryside setting Mountain peaks glowing in the sunset

Reference-guided



a cartoon animation of a castle in the distance a cartoon animation of a panda in the forest a professional photograph of a fire hydrant on the grass, ultra realistic a professional photograph of a tiger on the beach, ultra realistic



a cartoon animation of a squirrel in the forest a cartoon animation of a goat in the forest a cartoon animation of a panda in the forest a professional photograph of a puppy on the grass, ultra realistic

Image Variation



Futuristic holographic art with iridescent colors

Folk art style with decorative patterns and flat perspective



Traditional Chinese painting with fine ink brushwork and subtle gradients

Digital art in cyberpunk style with neon colors and high contrast

Prompt-guided

Ours

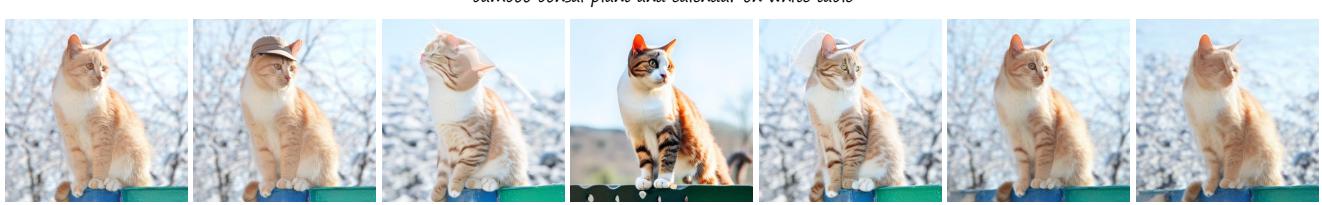
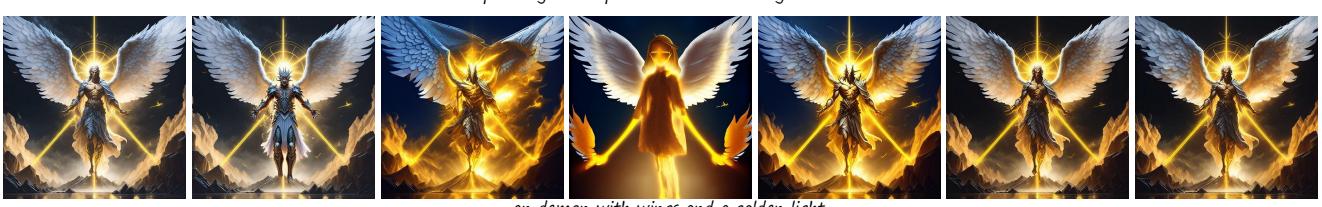
DDIM+PNP

DDIM+P2P

DI+PnP

DI+MasaCtrl

DDIM+MasaCtrl



Drag-guided



Ours



Drag Diffusion



FastDrag



RegionDrag



a photo of a cat



lighthouse, coast, grassland, sea



a photo of a leaf



a boy holding an umbrella



The oil painting of a beautiful scene



A photo of mountains covered with snow

Reference-guided



Ours



TF-ICON



PrimeComposer



DCCF



A professional photograph of a castle, ultra realistic



A cartoon animation of a panda in the forest



A professional photograph of a skyscraper in the distance, ultra realistic



An oil painting of a hamburger, Van Gogh Style



An oil painting of a chocolate doughnut, Van Gogh Style



An oil painting of a hot dog, Van Gogh Style