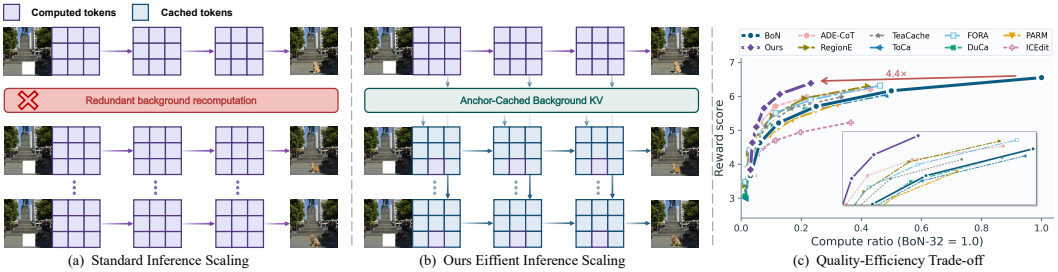
MASK-AWARE EFFICIENT INFERENCE-TIME SCALING FOR IMAGE INPAINTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Inference-time scaling has recently emerged as an effective paradigm for improving generation quality by allocating additional computation at test time. However, image inpainting remains largely unexplored in this setting, where complete-trajectory scaling is inefficient and structurally mismatched: it repeatedly scales the observed unmasked context and imposes a uniform computation budget on masked tokens with different refinement difficulties. Therefore, we identify two spatially non-uniform requirements for inpainting: inference-time scaling should exclude the observed unmasked region, while accounting for the varying difficulty across masked tokens. To this end, we propose a training-free, mask-aware framework for efficient inference-time scaling in image inpainting. For the unmasked context, Mask-Aware Initial Noise Construction and Anchor-Cached Background KV Cache remove observed-context redundancy by aligning background trajectories and reusing anchor background key/value states. For the masked region, Token-Aware Temporal Velocity Cache adapts masked-region updates by reusing velocity predictions for stabilized tokens and focusing computation on uncertain ones. Experiments on EditBench and BrushBench show that our method preserves scaling gains while reducing cost by up to $4.4\times$ and $4.6\times$ over standard scaling.



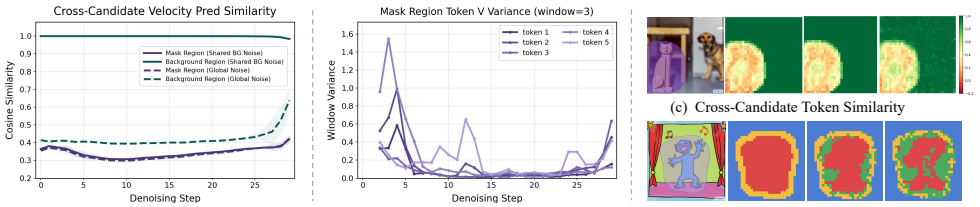
(a) Standard Inference Scaling (b) Ours Eiffient Inference Scaling (c) Quality-Efficiency Trade-off

Figure 1: **Comparison between standard inference scaling and our mask-aware efficient inference scaling.** (a) Standard inference scaling generates multiple full-image candidates independently, causing the computational cost to grow nearly linearly with the number of candidates. (b) Our method shares background evolution across candidates and concentrates computation on the masked region where meaningful diversity is needed. (c) By reducing redundant background computation, our method achieves a better quality–efficiency trade-off for image inpainting.

1 INTRODUCTION

Inference-time scaling has recently emerged as an effective paradigm for enhancing large language models, improving model output without retraining (Wei et al., 2022; Snell et al., 2024; Ma et al., 2025a). Inspired by this success, recent work has extended the inference-time scaling to generation models, such as text-to-image (Xie et al., 2025; Zhang et al., 2025a). These methods typically construct multiple candidate trajectories through noise perturbation, sampling, or search, and then select the best output using a reward or verifier model (Ma et al., 2025a; Zhang et al., 2025b). By expanding the search space, constructing multiple candidate trajectories improves generation quality, with inference cost growing nearly linearly with the number of candidates due to the full denoising process required for each candidate.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Background–Mask Region Comparison (b) Token variance within the Masked Region (c) Cross-Candidate Token Similarity (d) Convergence Dynamics of Tokens

Figure 2: **Motivation for mask-aware redundancy reduction in image inpainting.** (a) Unlike the masked region, the background should remain consistent across candidates, revealing substantial spatial redundancy. (b) Tokens within the masked region exhibit different temporal variances, indicating non-uniform denoising difficulty. (c) Shared background noise increases cross-candidate token similarity in preserved regions, enabling reliable background reuse. (d) As denoising proceeds, stabilized tokens show convergent velocity predictions and can be safely cached to reduce redundant computation.

Despite recent efforts to improve the cost–quality trade-off (Zhuo et al., 2025; Liu et al., 2025a), image inpainting task remains largely unexplored in visual inference-time scaling. A straightforward application of existing scaling strategies would operate at the level of full-image trajectories, as illustrated in Fig. 1(a). However, this global formulation is misaligned with the structure of inpainting, where only the masked (to be inpainted) region is generated, while the unmasked context should be preserved. Moreover, the binary mask treats all locations within the masked region uniformly, without distinguishing their potentially different generation difficulties. Consequently, existing complete-trajectory scaling introduces redundant computation over unchanged regions and imposes a uniform scaling budget within the mask, despite the spatially varying demands of inpainting.

We argue that *image inpainting requires spatially adaptive inference-time scaling*. This requirement is motivated by two observations. First, **the unmasked region should be excluded from inference-time scaling** for inpainting tasks: multiple candidates are intended to capture plausible alternatives for the missing content, whereas the unmasked region is observed context and should be excluded from scaling decisions, as shown in Fig. 1(b). Second, **refinement difficulty varies within the masked region**: different spatial tokens may stabilize at different rates during denoising, as illustrated in Fig. 2(b), and therefore may not require the same number of updates for inference-time scaling. Together, these observations suggest that inference-time scaling should avoid redundant computation over the observed context and instead focus on the more uncertain region to be inpainted.

To this end, we propose a training-free, mask-aware framework for efficient inference-time scaling in image inpainting. The framework directly targets the two spatially non-uniform demands above. To avoid redundant scaling over the unmasked region, we introduce a **Mask-Aware Initial Noise Construction** scheme, where all candidates share the same background noise while varying only the mask-region noise. This construction keeps the unmasked background aligned across candidates without suppressing meaningful diversity inside the masked region, as shown in Fig. 2(a) and (c). Based on this alignment, we further construct an **Anchor-Cached Background KV Cache**, which caches background key/value states from a complete anchor trajectory and reuses them as contextual memory for subsequent candidates. This avoids repeated computation over the unchanged background while preserving the global context required for coherent inpainting, as shown in Fig. 1(b). To adapt refinement within the mask, we introduce a **Token-Aware Temporal Velocity Cache**, which identifies masked tokens that have stabilized during denoising and directly reuses their cached predictions in subsequent steps, as shown in Fig. 2(d). In this way, computation is concentrated on masked tokens that remain uncertain or difficult to refine, rather than being uniformly spent across all masked tokens.

Our method can be integrated into standard inference-time scaling for image inpainting. Experiments on EditBench (Wang et al., 2023) and BrushBench (Ju et al., 2024) show that inference-time scaling improves inpainting quality, while our framework achieves these gains with substantially lower inference cost. Compared with vanilla trajectory-level scaling, our method reduces computational cost by up to $4.4\times$ on EditBench and $4.6\times$ on BrushBench, as Fig. 1(c).

Our main contributions are summarized as follows:

- To the best of our knowledge, we are the first to study inference-time scaling for image inpainting, revealing two forms of spatially adaptive demand: candidate-level scaling should

exclude the observed unmasked region, and refinement difficulty varies across masked-region tokens.

- We propose a training-free, mask-aware framework for inference-time scaling, which keeps candidate variation inside the masked region, while reducing redundant background computation across candidates and avoiding redundant updates of stabilized masked-region tokens.
- Extensive experiments on EditBench and BrushBench demonstrate that our framework retains the quality gains of inference-time scaling while substantially improving efficiency, reducing computational cost by up to $4.4\times$ and $4.6\times$, respectively, over standard scaling.

2 Related Work

Inference time scaling in image generation Inference-time scaling improves generation quality by allocating additional computation during sampling. A common strategy is noise scaling (Ma et al., 2025a; Xie et al., 2025), which samples multiple trajectories from perturbed noises and selects the best result, but its cost grows nearly linearly with the number of candidates. Recent methods reduce this cost-quality trade-off by enhancing diversity through prompt rewriting (Oh et al., 2025; Jiang et al., 2025) or reflection-based refinement (Li et al., 2025; Zhuo et al., 2025; Zhang et al., 2025a), formulating denoising as verifier-guided search (Ma et al., 2025a; Zhang et al., 2025b; Jain et al., 2025; He et al., 2025; Singhal et al., 2025; Ramesh & Mardani, 2025), or pruning low-potential trajectories with MLLM verifiers (Liu et al., 2025a; Zhang et al., 2025a; Oshima et al., 2025; Tian et al., 2025; Zhang et al., 2025c). However, most methods target open-ended text-to-image generation and allocate computation uniformly across tokens. In contrast, image inpainting requires diversity mainly within the masked region while preserving the known background.

Image Inpainting Image inpainting (Bertalmio et al., 2000) synthesizes mask regions while maintaining consistency with surrounding context. Recent progress is largely driven by diffusion models (Ho et al., 2020; Song et al., 2020) and flow-based models (Lipman et al., 2022; Yuan et al., 2023; Wan et al., 2025), enabling strong text-guided and structure-aware inpainting systems (Manukyan et al., 2023; Wang et al., 2025; Chen et al., 2024; Ma et al., 2024; 2025b). Early methods adapt pretrained generative or conditional diffusion models to inpainting (Avrahami et al., 2023; Rombach et al., 2022; Zhang et al., 2023), while later works improve controllability and context preservation with task-specific designs (Ju et al., 2024; Zhuang et al., 2024; Liu et al., 2024a; BlackForestLabs, 2024). Despite this progress, challenging inpainting scenarios remain sensitive to initial noise, especially when the masked content is semantically ambiguous or strongly coupled with the background.

Efficient Inference Time Scaling Efficient inference-time scaling commonly relies on early candidate pruning, where intermediate denoising states are scored to avoid full computation for unlikely trajectories. PARM (Zhang et al., 2025a) estimates both current clarity and future potential, VideoTTS (Liu et al., 2025a) guides generation with Tree-of-Frames and multiple verifiers, ICEdit (Zhang et al., 2025c) filters candidate noises after a few denoising steps, and ADE-CoT (Qu et al., 2026) improves early-stage assessment with edit-specific verification. These methods mainly optimize verifier reliability or search control during candidate selection. In contrast, our focus is not verifier design or search control, but computation allocation in the mask-guided inpainting search space.

3 Method

3.1 Overview

We study efficient inference-time scaling for image inpainting, where multiple candidates are sampled at inference time and a reward model selects the final output. We propose a training-free framework

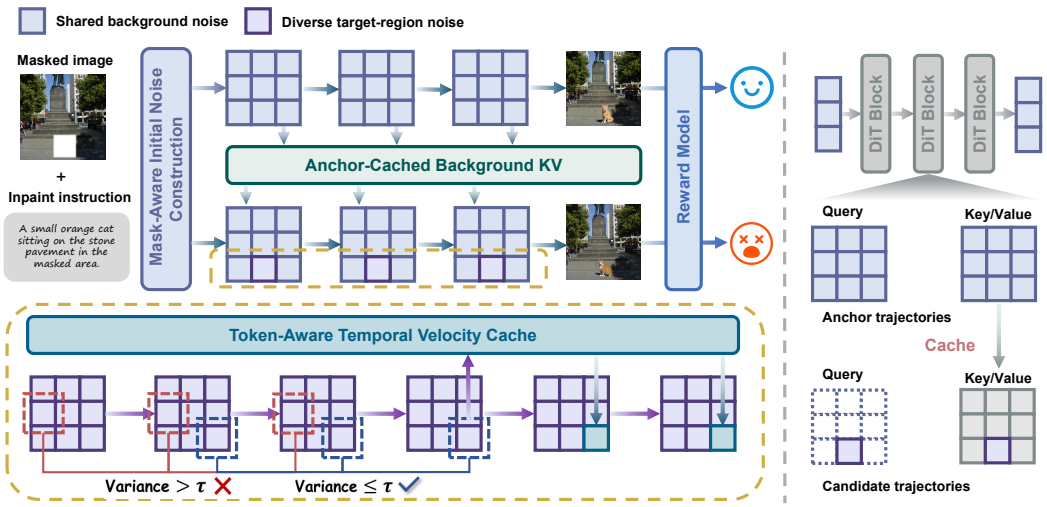


Figure 3: **Overall pipeline of our mask-aware efficient inference-time scaling framework.** Our method first constructs mask-aware initial noise by sharing background noise across candidates while varying only the target-region noise. It then builds an anchor trajectory to cache background K/V states for later candidates, and further applies Token-Aware temporal velocity caching to skip stabilized tokens during denoising.

with three tightly coupled components: *(i) mask-aware initial noise construction*, which aligns background evolution across candidates while preserving search diversity in the masked region; *(ii) anchor-cached background KV cache*, which preserves global context without recomputing stable background tokens for every candidate; and *(iii) a token-aware temporal velocity cache*, which allows stabilized tokens to exit active refinement early during denoising. The overall pipeline is illustrated in Fig. 3.

3.2 Mask-Aware Initial Noise Construction

Standard Best-of-N search resamples the entire initial noise independently for every candidate. Under deterministic sampling, the initial noise determines the entire denoising trajectory. Consequently, full-noise resampling forces the preserved background to follow different trajectories across candidates. For image inpainting, such background variation is redundant: it does not expand the meaningful candidate space, but still incurs additional denoising computation. Instead, we construct a mask-aware shared noise field across candidates, sharing the background noise while varying only the masked-region noise to concentrate diversity on the region that requires inpainting.

Let ϵ_{bg} denote the shared background noise and $\epsilon_{\text{tg}}^{(n)}$ denote the candidate-specific target-region noise for candidate n. The initial noise of candidate n is defined as

$$\epsilon^{(n)} = (1 - M) \odot \epsilon_{\text{bg}} + M \odot \epsilon_{\text{tg}}^{(n)}, \tag{1}$$

where $\odot$ denotes element-wise multiplication and the binary mask is lifted to the latent-token space. By sharing ϵ_{bg}, background tokens remain on aligned trajectories across candidates, while varying $\epsilon_{\text{tg}}^{(n)}$ preserves search diversity inside the masked region.

In practice, we first sample one candidate as an *anchor trajectory*. Subsequent candidates inherit the same background noise realization and differ only in the target-region initialization. This simple construction concentrates the search space on masked-region diversity while explicitly aligning background trajectories across candidates, thereby providing the foundation for reliable background KV reuse.

3.3 Anchor-Cached Background KV Cache

The shared background initialization aligns the preserved regions across candidates, but it does not imply that background tokens can be simply removed from the transformer. In image inpainting, the synthesized content must remain semantically coherent and geometrically compatible with its

surrounding context. Since attention layers mediate global token interactions, removing background tokens would deprive the dynamic region of essential context and bias its velocity prediction. Thus, background tokens should be retained as attention context, but need not be recomputed for every candidate. Tokens near the mask boundary strongly interact with the generated content and should remain dynamic, whereas tokens farther from the mask mostly preserve their appearance and geometry. We therefore partition image tokens into target tokens $\mathcal{T}$, boundary-band tokens $\mathcal{B}$, and background tokens $\mathcal{G}$. The dynamic region is defined as $\mathcal{D} = \mathcal{T} \cup \mathcal{B}$, while $\mathcal{G}$ is treated as reusable background context.

For the anchor trajectory, we perform the standard full forward pass at every denoising step. For each transformer block, we cache the key/value states of text tokens and stable background tokens. At step t, the cached memory is denoted by $\big(K^t_{\text{text}}, V^t_{\text{text}}, K^t_{\mathcal{G}}, V^t_{\mathcal{G}}\big)$. For each subsequent candidate, we recompute the transformer features only for the dynamic tokens $\mathcal{D}$. Let $(Q^t_{\mathcal{D}}, K^t_{\mathcal{D}}, V^t_{\mathcal{D}})$ denote their query, key, and value states. To avoid attention over a truncated sequence, we reconstruct the contextual memory by concatenating the anchor-cached text/background states with the current dynamic states:

$$K^t_{\text{all}} = [K^t_{\text{text}}; K^t_{\mathcal{G}}; K^t_{\mathcal{D}}], \quad V^t_{\text{all}} = [V^t_{\text{text}}; V^t_{\mathcal{G}}; V^t_{\mathcal{D}}]. \tag{2}$$

Attention is evaluated only for the dynamic queries:

$$\text{Attn}^t_{\mathcal{D}} = \text{Softmax}\left(\frac{Q^t_{\mathcal{D}} \left(K^t_{\text{all}}\right)^\top}{\sqrt{d}}\right) V^t_{\text{all}}. \tag{3}$$

In this way, dynamic tokens are updated with access to the same textual guidance and global visual context as in a full forward pass, while the stable background branch is not repeatedly evaluated. The background prediction is directly inherited from the anchor trajectory at the corresponding denoising step. Consequently, Computation is concentrated on the dynamic region that drives inpainting diversity.

3.4 Token-Aware Temporal Velocity Cache

Spatial reuse alone is not sufficient, because even within the dynamic region, denoising difficulty is highly non-uniform. Some tokens remain unstable for many steps and require continued refinement, while others quickly stabilize and no longer need to participate in every subsequent transformer update. Moreover, the safety of temporal reuse is also timestep-dependent: early denoising steps determine coarse structure and are more sensitive to stale predictions, whereas later steps mainly refine local details and can tolerate more aggressive reuse. To exploit this token- and timestep-dependent redundancy, we maintain a Token-Aware temporal velocity cache and progressively shrink the active token set during denoising.

For each candidate, let $\mathcal{A}_t \subseteq \mathcal{D}$ denote the active token set at step t. Only tokens in $\mathcal{A}_t$ participate in transformer computation, while inactive tokens reuse their latest cached prediction. Initially, all dynamic tokens are active, i.e., $\mathcal{A}_0 = \mathcal{D}$. After a short warm-up stage, we monitor the recent prediction history of each active token over a sliding window of W steps. For token i, let $v_i^{t-W+1:t}$ denote its history. We measure temporal stability with the coefficient of variation

$$\text{CV}^t_i = \frac{\text{Std}\left(v_i^{t-W+1:t}\right)}{\text{Mean}\left(|v_i^{t-W+1:t}|\right) + \epsilon}, \tag{4}$$

where the statistics are aggregated over the temporal window and channel dimension. We use a timestep-adaptive threshold τ_t to account for the changing sensitivity of the denoising process:

$$\tau_t = \tau_{\min} + \rho_t \left(\tau_{\max} - \tau_{\min}\right), \quad \rho_t = \frac{t - t_{\text{w}}}{T - t_{\text{w}}}, \tag{5}$$

where t_{w} denotes the warm-up length and ρ_t is clipped to $[0, 1]$. This schedule keeps reuse conservative in early steps and gradually relaxes the stability criterion as denoising proceeds. If $\text{CV}^t_i < \tau_t$, we regard token i as stabilized and remove it from the active set:

$$\mathcal{A}_{t+1} = \mathcal{A}_t \setminus \{i \mid \text{CV}^t_i < \tau_t\}. \tag{6}$$

Table 1: **Quantitative comparison with state-of-the-art image inpainting methods on BrushBench and EditBench.** The best results are shown in bold, and the second-best results are underlined.

Method	CLIP Score	HPS2.1	HPSv3	PickScore	VQA Score	ImageReward
BrushBench (Ju et al., 2024)						
SD 1.5 I (Rombach et al., 2022)	27.079	0.2718	6.6915	22.083	0.8821	1.1243
SD 2 I (Rombach et al., 2022)	26.887	0.2763	6.8049	22.163	0.8856	1.1750
DS-8 (Lykon, 2024)	27.231	0.2776	7.1643	22.319	0.8956	1.2353
CNI (Zhang et al., 2023)	26.700	0.2523	3.5653	21.569	0.8772	0.9640
BLD (Avrahami et al., 2023)	26.876	0.2679	5.3228	21.841	0.8978	1.1004
PowerPaint (Zhuang et al., 2024)	27.224	0.2728	6.5359	22.086	0.9029	1.2183
FreeInpaint (Gong et al., 2026)	27.279	0.2735	5.6266	21.990	0.8989	1.2003
PrefPaint (Liu et al., 2024b)	27.279	0.2675	6.1400	22.056	0.8953	1.1837
HD-Painter (Manukyan et al., 2023)	27.043	0.2761	6.7885	22.081	0.8948	1.1771
ASUKA (Wang et al., 2025)	26.964	0.2752	6.9892	22.292	0.9067	1.2703
BrushNet (Ju et al., 2024)	27.170	0.2715	6.7593	22.127	0.8942	1.2176
FLUX.1 Fill (BlackForestLabs, 2024)	27.765	0.2757	6.7501	22.296	0.9047	1.2618
Ours	**28.412**	**0.2841**	**7.6706**	**22.616**	**0.9269**	**1.4172**
EditBench (Wang et al., 2023)						
SD 1.5 I (Rombach et al., 2022)	26.740	0.2376	2.3967	20.920	0.6760	-0.0837
SD 2 I (Rombach et al., 2022)	26.795	0.2436	3.0521	21.109	0.7277	0.0456
DS-8 (Lykon, 2024)	27.192	0.2425	3.1740	21.131	0.7033	0.0869
CNI (Zhang et al., 2023)	26.416	0.2214	-0.7206	20.532	0.6666	-0.3368
BLD (Avrahami et al., 2023)	27.765	0.2442	2.3655	21.037	0.7670	0.2255
PowerPaint (Zhuang et al., 2024)	27.450	0.2400	2.5931	20.939	0.7395	0.1102
FreeInpaint (Gong et al., 2026)	27.213	0.2311	1.2944	20.709	0.7121	-0.0819
PrefPaint (Liu et al., 2024b)	26.421	0.2272	0.7661	20.776	0.6572	-0.2634
HD-Painter (Manukyan et al., 2023)	27.121	0.2451	3.0199	21.106	0.7496	0.1690
ASUKA (Wang et al., 2025)	27.712	0.2455	4.1269	21.347	0.8164	0.4882
BrushNet (Ju et al., 2024)	27.818	0.2417	2.9671	21.098	0.7454	0.1479
FLUX.1 Fill (BlackForestLabs, 2024)	28.757	0.2460	3.6644	21.313	0.8275	0.4827
Ours	**30.078**	**0.2588**	**5.5138**	**21.747**	**0.8772**	**0.8361**

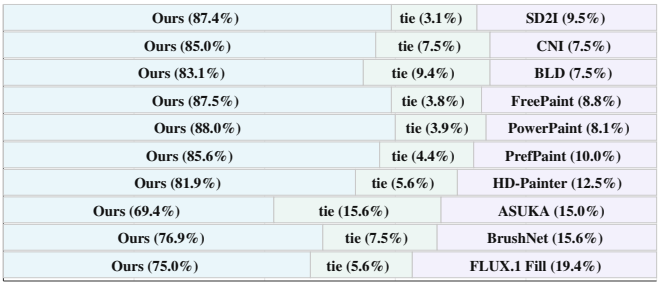


Figure 4: **User study results for our method.**

In later denoising steps, only the remaining active tokens are recomputed. Stabilized tokens directly use their cached velocity from the recent active step, while background tokens continue to use the anchor-cached branch. This yields a temporally adaptive refinement process: easy tokens exit active refinement early, difficult tokens remain active, and the reused token coverage naturally expands over time. As a result, the effective computation is governed by the shrinking active set rather than the original mask size, mitigating the dependence of inference cost on masked regions.

4 Experiments

4.1 Experimental Setup

Implementation details. We instantiate our framework on the transformer-based FLUX.1 Fill model (BlackForestLabs, 2024) and evaluate it under multi-candidate inpainting search. Unless otherwise specified, each input is sampled with $N = 32$ candidates, 30 denoising steps, and a guidance scale of 30.0 at 512×512 resolution. For the anchor-cached background KV module, we set the boundary band to 2 latent-token units. For Token-Aware temporal reuse, we use a timestep-



Figure 5: **Qualitative comparisons with state-of-the-art image inpainting models.**

Table 2: **Comparison with efficient inference-time scaling and cache-based acceleration methods on BrushBench and EditBench.** The cost ratio is normalized to vanilla Best-of-N inference; lower is better. The best results are shown in bold, and the second-best results are underlined.

Method	CLIP Score↑	HPS 2.1 ↑	HPS $v3$ ↑	Pick Score↑	VQA Score↑	Image Reward↑	Cost Ratio↓
BrushBench (Ju et al., 2024)							
BoN (Ma et al., 2025a)	**28.420**	**0.2847**	**7.7924**	22.608	0.9265	1.4069	1.0000
PARM (Zhang et al., 2025a)	28.202	0.2830	7.4523	22.548	0.9198	1.3984	0.4302
ICEdit (Zhang et al., 2025c)	28.002	0.2799	7.2162	22.448	0.9157	1.3622	0.3865
ADE-CoT (Qu et al., 2026)	28.304	0.2839	7.6383	22.582	0.9228	1.4022	0.4120
DUCA (Zou et al., 2024b)	28.327	0.2835	7.5452	22.575	0.9249	1.3997	0.4280
FORA (Selvaraju et al., 2024)	28.316	0.2842	7.6620	22.576	0.9234	1.4145	0.4620
TeaCache (Liu et al., 2025b)	28.313	0.2830	7.5816	22.570	0.9235	1.3989	0.3340
ToCa (Zou et al., 2024a)	28.299	0.2836	7.6088	22.574	0.9261	1.4026	0.4800
RegionE (Chen et al., 2026)	28.381	0.2844	7.7664	22.600	0.9250	1.4124	0.4100
Ours	28.412	0.2841	7.6706	**22.616**	**0.9269**	**1.4172**	**0.2140**
EditBench (Wang et al., 2023)							
BoN (Ma et al., 2025a)	**30.187**	**0.2605**	**5.6922**	**21.793**	0.8771	**0.8631**	1.0000
PARM (Zhang et al., 2025a)	29.850	0.2592	5.4689	21.720	0.8576	0.8236	0.4302
ICEdit (Zhang et al., 2025c)	29.454	0.2557	5.0291	21.637	0.8511	0.7623	0.3865
ADE-CoT (Qu et al., 2026)	30.041	0.2587	5.5021	21.729	0.8635	0.8302	0.4120
DUCA (Zou et al., 2024b)	29.988	0.2575	5.4329	21.684	0.8696	0.8029	0.4280
FORA (Selvaraju et al., 2024)	30.032	0.2566	5.4663	21.620	0.8665	0.8321	0.4620
TeaCache (Liu et al., 2025b)	30.094	0.2574	5.2583	21.721	0.8706	0.7915	0.3330
ToCa (Zou et al., 2024a)	30.061	0.2572	5.4452	21.681	0.8694	0.8143	0.4800
RegionE (Chen et al., 2026)	30.086	0.2581	5.4463	21.729	0.8746	0.8242	0.4220
Ours	30.078	0.2588	5.5138	21.747	**0.8772**	0.8361	**0.2290**

adaptive coefficient-of-variation threshold, with $\tau_{\min} = 0.5$ and $\tau_{\max} = 2.0$, unless otherwise stated. All experiments are conducted on NVIDIA A6000 GPUs.

Evaluation settings. We evaluate on two widely used image inpainting benchmarks, EditBench (Wang et al., 2023) and BrushBench (Ju et al., 2024). To assess generation quality from complementary perspectives, we report CLIP Score (Hessel et al., 2021), HPS2.1 (Wu et al., 2023), HPSv3 (Ma et al., 2025c), PickScore (Kirstain et al., 2023), VQA Score (Lin et al., 2024), and ImageReward (Xu et al., 2023). These metrics jointly measure text–image alignment, human preference, visual plausibility, and instruction satisfaction. For efficiency, we report a FLOP-based cost ratio normalized to vanilla multi-candidate Best-of-N inference, which recomputes the full transformer for every candidate. Unless otherwise noted, for every method, we select the top-1 candidate from (N=32) using a fixed composite reward that equally averages the z-score-normalized CLIPScore, HPSv3, and VQA Score. The remaining metrics are used only for evaluation.

Table 3: **Effects of the Proposed Components on Efficiency and Performance.** Colored rows highlight the complete configuration for each ablation group: mask-aware initial noise construction, spatial background K/V reuse, and temporal token reuse. MIN denotes mask-aware initial noise; BG-KV denotes anchor-cached background key/value reuse; GVC denotes global velocity cache; TVC denotes Token-Aware temporal velocity cache.

Variant	CLIP Score↑	HPS2.1↑	HPSv3↑	PickScore↑	VQA Score↑	ImageReward↑	Cost Ratio↓
a) Global BoN	30.19	0.261	5.69	21.79	0.877	0.863	1.000
b) + MIN	30.05	0.260	5.81	21.76	0.875	0.847	1.000
c) + BG-KV w/o MIN	29.03	0.246	3.49	21.33	0.844	0.525	0.279
d) + MIN + Vel	29.74	0.241	1.81	21.17	0.827	0.413	0.279
e) + MIN + BG-KV	30.05	0.259	5.59	21.74	0.875	0.833	0.279
f) + MIN + BG-KV + GVC	29.87	0.255	5.06	21.70	0.864	0.779	0.234
g) + MIN + BG-KV + TVC	30.08	0.259	5.51	21.75	0.877	0.836	0.229

4.2 Comparison with SOTA Inpainting Methods

We first compare our method with representative state-of-the-art image inpainting models (Rombach et al., 2022; Zhang et al., 2023; Avrahami et al., 2023; Zhuang et al., 2024; Ju et al., 2024; Liu et al., 2024b;a; BlackForestLabs, 2024; Manukyan et al., 2023; Wang et al., 2025). As summarized in Tab 1, our method consistently achieves the best performance across all evaluated metrics on both EditBench and BrushBench. These results indicate that the proposed inference-time scaling strategy improves prompt alignment and human-preference quality while preserving strong perceptual fidelity.

Compared with FLUX.1 Fill, the base model before applying inference-time scaling, our method substantially improves all metrics on EditBench and BrushBench. This verifies that multi-candidate inference-time scaling can improve a fixed pretrained inpainting generator. The qualitative comparisons in Fig. 5 further demonstrate that, especially in challenging cases involving text rendering, material and identity preservation, and counterfactual composition, competing methods often suffer from incomplete prompt grounding or local artifacts, whereas our method accurately renders textual content, preserves materials and object identities without confusion, and produces more plausible and context-consistent completions. We further conduct a user study, as shown in Fig. 4, to evaluate human judgments of instruction following and visual aesthetics. Our method receives consistently higher preference rates, indicating that its inpainting results are better aligned with human preference.

4.3 Comparison with Efficient Inference-Time Scaling Methods

We further compare our method with representative acceleration strategies for inference-time scaling and diffusion transformer inference. The baselines include pruning-based candidate selection methods, such as PARM (Zhang et al., 2025a), ICEdit (Zhang et al., 2025c) and ADE-CoT (Qu et al., 2026), as well as cache- or reuse-based acceleration methods, including DUCA (Zou et al., 2024b), FORA (Selvaraju et al., 2024), TeaCache (Liu et al., 2025b), ToCa (Zou et al., 2024a), and RegionE (Chen et al., 2026). All methods are evaluated on BrushBench and EditBench under the same multi-candidate inpainting setting.

Tab 2 reports both generation quality and normalized cost ratio with respect to vanilla BoN. BoN provides a strong quality reference, but requires full-image recomputation for all candidates. Existing acceleration methods substantially reduce the cost ratio, yet often incur measurable degradation in alignment or human-preference metrics. In contrast, our method achieves the strongest overall quality–efficiency trade-off among the accelerated variants. It achieves the lowest normalized cost ratios on both benchmarks, 0.214 on BrushBench and 0.229 on EditBench, while maintaining near-BoN generation quality and delivering the strongest overall performance among accelerated variants. This indicates that exploiting the mask-local diversity of inpainting is more effective than directly applying generic pruning or full-image cache reuse. As shown in Fig. 6, our method consistently approaches the upper-left quality–efficiency frontier across metrics under different candidate budgets, achieving near-BoN performance with substantially lower inference cost than pruning or cache-based acceleration methods.

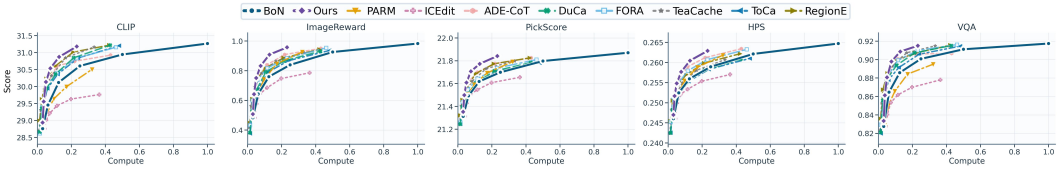


Figure 6: **Quality–efficiency trade-offs under different candidate budgets.**



Figure 7: **Qualitative ablation on the proposed reuse mechanisms.**

4.4 Ablation on Spatial and Temporal Reuse

We conduct ablation studies to isolate the contribution of each component in our mask-aware scaling framework. The quantitative results are summarized in Tab. 3, with the corresponding qualitative comparisons shown in Fig. 7.

Mask-aware noise construction and anchor-cached background KV. Rows a) and b) show that replacing globally independent initial noise with our mask-aware initial noise construction yields comparable inpainting quality. This confirms that sharing background noise does not reduce the effectiveness of candidate search, since meaningful diversity is primarily expressed in the masked region. Rows b) and e) further show that anchor-cached background KV substantially improves efficiency without sacrificing generation quality. Rows c) and e) demonstrate that this reuse must be built upon mask-aware initial noise construction. Without trajectory alignment, directly substituting cached background states leads to severe quality degradation, indicating that cross-candidate KV reuse requires aligned background evolution. As illustrated in Fig. 7, BG-KV reuse without mask-aware initialization introduces visible structural artifacts and contextual inconsistencies, whereas combining MIN with BG-KV produces coherent completions. Rows d) and e) further demonstrate the necessity of reusing background K/V states rather than only carrying over background velocity predictions. Velocity-only reuse weakens the interaction between the inpainted region and its surrounding context, resulting in less faithful object–context integration.

Token-Aware temporal velocity cache. Rows f) and g) show that temporal caching further improves inference efficiency after spatial redundancy has been reduced. Compared with global velocity reuse, our token- and timestep-aware strategy achieves a lower cost ratio with better overall quality, because it preserves computation for tokens that remain difficult while freezing only stabilized ones. The qualitative comparisons in Fig. 7 further show that global velocity reuse can degrade locally difficult regions, whereas Token-Aware temporal reuse better preserves object structure and contextual details.

5 Conclusion

We presented a mask-aware inference-time scaling framework for efficient image inpainting. The method is built on two observations: background tokens remain highly stable across candidates and can reuse exact cached K/V states, while many dynamic-region tokens become temporally stable during denoising and support Token-Aware velocity reuse. By combining these two mechanisms, our approach reduces both cross-candidate spatial redundancy and within-candidate temporal redundancy without requiring additional training. Experiments on EditBench and BrushBench show that the proposed design achieves a favorable efficiency-quality trade-off, substantially reducing inference cost while preserving competitive search performance. We believe this work highlights a useful perspective for future research on inference-time scaling in image inpainting: Computation should be allocated selectively to meaningful regions and moments, rather than applied uniformly.

AI USE STATEMENT

Generative AI tools were used to polish the manuscript's writing and to assist with LaTeX template migration and typesetting, including figure, table, and reference formatting.

REFERENCES

Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM transactions on graphics*, 2023.

Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000.

BlackForestLabs. Flux.1-fill-dev. https://huggingface.co/black-forest-labs/FLUX.1-Fill-dev, 2024. Accessed: 2025-07-12.

Pengtao Chen, Xianfang Zeng, Maosen Zhao, Mingzhu Shen, Peng Ye, Bangyin Xiang, Zhibo Wang, Wei Cheng, Gang Yu, and Tao Chen. Regione: Adaptive region-aware generation for efficient image editing. In *ICLR*, 2026.

Qihua Chen, Yue Ma, Hongfa Wang, Junkun Yuan, Wenzhe Zhao, Qi Tian, Hongmei Wang, Shaobo Min, Qifeng Chen, and Wei Liu. Follow-your-canvas: Higher-resolution video outpainting with extensive content generation. *arXiv preprint arXiv:2409.01055*, 2024.

Chao Gong, Dong Li, Yingwei Pan, Jingjing Chen, Ting Yao, and Tao Mei. Freeinpaint: Tuning-free prompt alignment and visual rationality enhancement in image inpainting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4239–4247, 2026. doi: 10.1609/aaai.v40i6.42420.

Haoran He, Jiajun Liang, Xintao Wang, Pengfei Wan, Di Zhang, Kun Gai, and Ling Pan. Scaling image and video generation via test-time evolutionary search. *CoRR*, abs/2505.17618, 2025.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *EMNLP*, pp. 7514–7528, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 2020.

Vineet Jain, Kusha Sareen, Mohammad Pedramfar, and Siamak Ravanbakhsh. Diffusion tree sampling: Scalable inference-time alignment of diffusion models. *CoRR*, abs/2506.20701, 2025.

Dongzhi Jiang, Ziyu Guo, Renrui Zhang, Zhuofan Zong, Hao Li, Le Zhuo, Shilin Yan, Pheng-Ann Heng, and Hongsheng Li. T2I-R1: reinforcing image generation with collaborative semantic-level and token-level cot. *CoRR*, abs/2505.00703, 2025.

Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. In *European Conference on Computer Vision*, 2024.

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*, 2023.

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. *CoRR*, abs/2506.15742, 2025.

Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Arsh Koneru, Yusuke Kato, Kazuki Kozuka, and Aditya Grover. Reflect-dit: Inference-time scaling for text-to-image diffusion transformers via in-context reflection. *CoRR*, abs/2503.12271, 2025.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Fangfu Liu, Hanyang Wang, Yimo Cai, Kaiyan Zhang, Xiaohang Zhan, and Yueqi Duan. Video-t1: Test-time scaling for video generation. *CoRR*, abs/2503.18942, 2025a.

Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7353–7363, 2025b.

Haipeng Liu, Yang Wang, Biao Qian, Meng Wang, and Yong Rui. Structure matters: Tackling the semantic discrepancy in diffusion models for image inpainting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024a.

Kendong Liu, Zhiyu Zhu, Chuanhao Li, Hui Liu, Huanqiang Zeng, and Junhui Hou. Prefpaint: Aligning image inpainting diffusion model with human preference. *Advances in Neural Information Processing Systems*, 2024b.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, Guopeng Li, Yuang Peng, Quan Sun, Jingwei Wu, Yan Cai, Zheng Ge, Ranchen Ming, Lei Xia, Xianfang Zeng, Yibo Zhu, Binxing Jiao, Xiangyu Zhang, Gang Yu, and Daxin Jiang. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025c.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Lykon. Dreamshaper 8 inpainting. https://huggingface.co/Lykon/dreamshaper-8-inpainting, 2024. Model card, accessed 2026-09-25.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi S. Jaakkola, Xuhui Jia, and Saining Xie. Scaling inference time compute for diffusion models. In *CVPR*, pp. 2523–2534, 2025a.

Yue Ma, Hongyu Liu, Hongfa Wang, Heng Pan, Yingqing He, Junkun Yuan, Ailing Zeng, Chengfei Cai, Heung-Yeung Shum, Wei Liu, et al. Follow-your-emoji: Fine-controllable and expressive freestyle portrait animation. In *SIGGRAPH Asia 2024 Conference Papers*, 2024.

Yue Ma, Zexuan Yan, Hongyu Liu, Hongfa Wang, Heng Pan, Yingqing He, Junkun Yuan, Ailing Zeng, Chengfei Cai, Heung-Yeung Shum, et al. Follow-your-emoji-faster: Towards efficient, fine-controllable, and expressive freestyle portrait animation. *arXiv preprint arXiv:2509.16630*, 2025b.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. *arXiv preprint arXiv:2508.03789*, 2025c.

Hayk Manukyan, Andranik Sargsyan, Barsegh Atanyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Hd-painter: high-resolution and prompt-faithful text-guided image inpainting with diffusion models. In *International Conference on Learning Representations*, 2023.

Yoonjin Oh, Yongjin Kim, Hyomin Kim, Donghwan Chi, and Sungwoong Kim. Object-centric self-improving preference optimization for text-to-image generation. *CoRR*, abs/2506.02015, 2025.

Yuta Oshima, Masahiro Suzuki, Yutaka Matsuo, and Hiroki Furuta. Inference-time text-to-video alignment with diffusion latent beam search. *CoRR*, abs/2501.19252, 2025.

Xiangyan Qu, Zhenlong Yuan, Jing Tang, Rui Chen, Datao Tang, Meng Yu, Lei Sun, Yancheng Bai, Xiangxiang Chu, Gaopeng Gou, et al. From scale to speed: Adaptive test-time scaling for image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23272–23282, 2026.

Vignav Ramesh and Morteza Mardani. Test-time scaling of diffusion models via noise trajectory search. *CoRR*, abs/2506.03164, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Pratheba Selvaraju, Tianyu Ding, Tianyi Chen, Ilya Zharkov, and Luming Liang. Fora: Fast-forward caching in diffusion transformer acceleration. *arXiv preprint arXiv:2407.01425*, 2024.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. *CoRR*, abs/2501.06848, 2025.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *CoRR*, abs/2408.03314, 2024.

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

Rui Tian, Mingfei Gao, Mingze Xu, Jiaming Hu, Jiasen Lu, Zuxuan Wu, Yinfei Yang, and Afshin Dehghan. Unigen: Enhanced training & test-time strategies for unified multimodal understanding and generation. *CoRR*, abs/2505.14682, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Su Wang, Chitwan Saharia, Ceslee Montgomery, Jordi Pont-Tuset, Shai Noy, Stefano Pellegrini, Yasumasa Onoe, Sarah Laszlo, David J Fleet, Radu Soricut, et al. Imagen editor and editbench: Advancing and evaluating text-guided image inpainting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Yikai Wang, Chenjie Cao, Junqiu Yu, Ke Fan, Xiangyang Xue, and Yanwei Fu. Towards enhanced image inpainting: Mitigating unwanted object insertion and preserving color consistency. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

Enze Xie, Junsong Chen, Yuyang Zhao, Jincheng Yu, Ligeng Zhu, Yujun Lin, Zhekai Zhang, Muyang Li, Junyu Chen, Han Cai, Bingchen Liu, Daquan Zhou, and Song Han. SANA 1.5: Efficient scaling of training-time and inference-time compute in linear diffusion transformer. *CoRR*, abs/2501.18427, 2025.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 2023.

Jinyan Ye, Zhongjie Duan, Zhiwen Li, Cen Chen, Daoyuan Chen, Yaliang Li, and Yingda Chen. Spectral evolution search: Efficient inference-time scaling for reward-aligned image generation. *CoRR*, abs/2602.03208, 2026.

Junkun Yuan, Xinyu Zhang, Hao Zhou, Jian Wang, Zhongwei Qiu, Zhiyin Shao, Shaofeng Zhang, Sifan Long, Kun Kuang, Kun Yao, et al. Hap: Structure-aware masked image modeling for human-centric perception. *Advances in Neural Information Processing Systems*, 2023.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Renrui Zhang, Chengzhuo Tong, Zhizheng Zhao, Ziyu Guo, Haoquan Zhang, Manyuan Zhang, Jiaming Liu, Peng Gao, and Hongsheng Li. Let's verify and reinforce image generation step by step. In *CVPR*, pp. 28662–28672, 2025a.

Xiangcheng Zhang, Haowei Lin, Haotian Ye, James Y. Zou, Jianzhu Ma, Yitao Liang, and Yilun Du. Inference-time scaling of diffusion models through classical search. *CoRR*, abs/2505.23614, 2025b.

Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. In-context edit: Enabling instructional image editing with in-context generation in large scale diffusion transformer. *CoRR*, abs/2504.20690, 2025c.

Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. In *European Conference on Computer Vision*, 2024.

Le Zhuo, Liangbing Zhao, Sayak Paul, Yue Liao, Renrui Zhang, Yi Xin, Peng Gao, Mohamed Elhoseiny, and Hongsheng Li. From reflection to perfection: Scaling inference-time optimization for text-to-image diffusion models via reflection tuning. *CoRR*, abs/2504.16080, 2025.

Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. *arXiv preprint arXiv:2410.05317*, 2024a.

Chang Zou, Evelyn Zhang, Runlin Guo, Haohang Xu, Conghui He, Xuming Hu, and Linfeng Zhang. Accelerating diffusion transformers with dual feature caching. *arXiv preprint arXiv:2412.18911*, 2024b.

A PRELIMINARIES

A.1 FLOW-BASED MODEL AND IMAGE INPAINTING

Modern image generation and editing systems increasingly adopt flow matching and rectified-flow formulations (Lipman et al., 2023; Liu et al., 2022; Chen et al., 2026; Ye et al., 2026). In this framework, the generator learns a time-dependent velocity field v_θ that transports a simple noise prior to the target image distribution through an ordinary differential equation (ODE). Given conditioning variables, the latent trajectory satisfies

$$\frac{dx_t}{dt} = v_\theta(x_t, t; \mathcal{C}), \qquad x_T \sim \mathcal{N}(0, I), \tag{7}$$

where $\mathcal{C}$ denotes the conditioning information. In rectified flow, the transport is further simplified by assuming an approximately straight trajectory between clean data and noise, which yields the familiar linear interpolation form used in modern flow-based solvers (Liu et al., 2022; Chen et al., 2026). In practice, the ODE is discretized into iterative updates of the form

$$x_{t-\Delta t} = x_t - \Delta t\, v_\theta(x_t, t; \mathcal{C}). \tag{8}$$

For image inpainting, we instantiate $\mathcal{C}$ with a source image I_{src}, a binary mask M, and an editing instruction c. The denoising process progressively transforms the initial noise x_T into an edited image that modifies the masked region while preserving the unmasked context.

Under deterministic ODE sampling, the full generation process defines a deterministic flow map from the initial noise space to the output image space (Ye et al., 2026). We denote this map by

$$I = \Psi_\theta(x_T; I_{\text{src}}, M, c). \tag{9}$$

This perspective is particularly useful for image inpainting: once the source image, mask, and instruction are fixed, the final output is uniquely determined by the initial noise. Consequently, the initial noise becomes the primary degree of freedom for inference-time exploration. In practice, recent inpainting systems such as Step1X-Edit and FLUX Kontext follow this flow-based paradigm and estimate the velocity field with a transformer backbone over text and image tokens (Liu et al., 2025c; Labs et al., 2025).

A.2 INFERENCE-TIME SCALING

Given a source image I_{src}, a binary mask M, and a text prompt c, the goal of image inpainting is to generate an image I whose masked region is semantically coherent with c while the unmasked region remains consistent with the observed context in I_{src}. In inference-time scaling, the standard Best-of-N paradigm consists of two stages. (1) **Generation**: a set of N candidate images $\{I^{(1)}, I^{(2)}, \ldots, I^{(N)}\}$ is generated by varying the initial noise while keeping the same source image, mask, and prompt. (2) **Selection**: a verifier assigns a scalar score to each candidate, reflecting its inpainting quality and semantic alignment with the prompt,

$$R : I_{\text{src}} \times M \times I \times c \to \mathbb{R}. \tag{10}$$

The final output is selected as

$$I^\star = \arg\max_{I^{(n)}} R(I_{\text{src}}, M, I^{(n)}, c). \tag{11}$$

From this perspective, inference-time scaling can be viewed as a search process over candidate trajectories induced by different initial noises.

B ADDITIONAL INFERENCE-TIME SCALING RESULTS

Inference-time scaling with individual reward models. Table 4 demonstrates a consistent scaling trend that is independent of the composite reward. For every individual reward model, the directly optimized metric increases strictly monotonically as the candidate budget grows from $N = 1$ to $N = 96$ under both BoN and Ours. Under BoN, CLIP Score increases from 28.757 to 31.565, while HPSv3 increases from 3.6644 to 6.7233. Under Ours, CLIP Score increases from 28.757 to 31.428,

Table 4: **Single-reward inference-time scaling on EditBench.** Each row uses one evaluator as the sole top-1 selection reward. $N = 1$ is the single-candidate baseline. The directly optimized full-image score increases strictly with budget for every row.

Setting	Reward model	$N = 1$	$N = 8$	$N = 16$	$N = 32$	$N = 64$	$N = 96$
BoN	CLIP Score	28.757	30.605	30.944	31.066	31.398	**31.565**
	HPS	0.2460	0.2589	0.2611	0.2637	0.2644	**0.2660**
	HPSv3	3.6644	5.7025	6.0609	6.1994	6.5984	**6.7233**
	PickScore	21.312	21.699	21.797	21.860	21.904	**21.946**
	VQA Score	0.8275	0.8908	0.9112	0.9134	0.9213	**0.9259**
	ImageReward	0.4827	0.8394	0.9162	0.9532	1.0085	**1.0363**
Ours	CLIP Score	28.757	30.570	30.704	30.980	31.272	**31.428**
	HPS	0.2460	0.2568	0.2596	0.2630	0.2635	**0.2644**
	HPSv3	3.6644	5.6368	5.9333	6.1528	6.5333	**6.6805**
	PickScore	21.312	21.672	21.778	21.836	21.893	**21.923**
	VQA Score	0.8275	0.8995	0.9052	0.9116	0.9196	**0.9238**
	ImageReward	0.4827	0.8231	0.9165	0.9333	0.9910	**1.0196**

Table 5: **Cross-model evaluation of single-reward-selected candidates at $N = 32$.** Rows name the selection reward; columns score the same selected output. Bold marks the directly optimized metric. Every entry exceeds its setting's single-candidate baseline.

Setting	Selection reward	CLIP	HPS	HPSv3	PickScore	VQA	ImageReward
	Baseline ($N = 1$)	28.757	0.2460	3.6644	21.312	0.8275	0.4827
BoN	CLIP Score	**31.066**	0.2500	4.1039	21.473	0.8602	0.5974
	HPS	29.629	**0.2637**	5.4056	21.739	0.8489	0.7386
	HPSv3	29.122	0.2566	**6.1994**	21.584	0.8375	0.6821
	PickScore	29.533	0.2586	5.2268	**21.860**	0.8502	0.7328
	VQA Score	29.443	0.2502	4.1302	21.434	**0.9134**	0.6262
	ImageReward	29.558	0.2548	4.7628	21.562	0.8588	**1.0085**
	Baseline ($N = 1$)	28.757	0.2460	3.6644	21.312	0.8275	0.4827
Ours	CLIP Score	**30.980**	0.2494	3.9573	21.472	0.8642	0.6307
	HPS	29.590	**0.2630**	5.1612	21.724	0.8421	0.7235
	HPSv3	29.192	0.2551	**6.1528**	21.585	0.8419	0.6670
	PickScore	29.439	0.2572	4.9652	**21.836**	0.8523	0.7090
	VQA Score	29.545	0.2493	4.1058	21.433	**0.9116**	0.6321
	ImageReward	29.488	0.2531	4.6932	21.535	0.8633	**0.9333**

and HPSv3 increases from 3.6644 to 6.6805. HPS, PickScore, VQA Score, and ImageReward exhibit the same monotonic trend, showing that the scaling behavior consistently holds across all six reward models rather than being induced by a particular composite selector.

Importantly, the scaling curves of Ours remain close to those of BoN across candidate budgets. At $N = 96$, the differences between Ours and BoN are only 0.137 for CLIP Score and 0.0428 for HPSv3. Similarly small gaps are observed for the other directly optimized metrics. At some intermediate budgets, Ours even matches or slightly outperforms BoN on VQA Score or ImageReward. These results indicate that, although Ours reuses anchor-cached background states, it largely preserves the candidate-search capability and the quality gains obtained by increasing the inference-time budget under full-trajectory BoN.

Table 5 further examines whether these improvements result only from over-optimizing the selection reward. At $N = 32$, selecting candidates with any individual reward improves not only the directly optimized metric but also all five held-out evaluation metrics over the $N = 1$ baseline for both BoN and Ours.

Held-out metric curves need not be strictly monotonic because different evaluators may rank the same candidate pool differently. Nevertheless, the consistently positive cross-model gains show that inference-time scaling improves overall candidate quality rather than merely exploiting the reward model used for selection. The similar cross-evaluator performance of Ours and BoN further demon-

Table 6: **Exact configuration of the reported token-aware temporal velocity-cache result.**

Item	Reported TVC configuration
Backbone	FLUX.1 Fill-dev
Resolution / steps	512×512 / 30
Guidance / strength	30.0 / 1.0
Candidate batch size	4
Boundary dilation	2 image-token units
CV base / window / warm-up	0.5 / 3 / 5
Numerical precision	bfloat16

strates that background-state reuse preserves the main search gains and cross-reward generalization of standard BON while avoiding redundant background computation.

C Implementation and Reproducibility Details

Image-token partition. At 512×512 resolution, the FLUX.1 Fill VAE produces a 64×64 latent grid. FLUX packs each 2×2 latent neighborhood into one transformer token, yielding a 32×32 grid and 1,024 image tokens. We resize the binary inpainting mask to the latent grid with nearest-neighbor interpolation, pack it in the same way, and classify a packed token as a mask token when its mean mask value is larger than 0.5. Let $\mathcal{M}$ denote these mask tokens. We dilate $\mathcal{M}$ by two token units with a 5×5 max-pooling kernel. The dilated ring outside the mask is the boundary set $\mathcal{B}$, and the remaining image tokens form the far-background set $\mathcal{F}$. The dynamic set is therefore $\mathcal{D} = \mathcal{M} \cup \mathcal{B}$, while $\mathcal{F}$ is reused across candidates.

Noise construction and anchor cache. For one input, we sample one background noise tensor and use it for the complete anchor initialization. Every reuse candidate keeps this same noise outside the pixel mask and samples an independent target noise tensor inside the mask. This aligns the background trajectories without forcing identical masked content. The anchor performs a full transformer forward at every denoising step. At each double- and single-stream block, it stores the text and far-background key/value tensors; it also stores the final far-background velocity prediction for that step. The benchmark implementation moves these per-step caches to CPU between uses. In the evaluated implementation, cache construction invokes explicit text/far-background QKV projections before ordinary attention; C_{anchor} includes this overhead and does not credit fused K/V extraction. Logically, a reuse candidate updates only its active dynamic tokens; the batched kernel evaluates the group union detailed below. Its attention context concatenates cached anchor text K/V, cached anchor far-background K/V, and group-union dynamic K/V. Dynamic positions outside that union are omitted from attention and reuse their last velocity, while the far background uses the step-aligned anchor prediction.

Exact token-convergence statistic. Let $v_{b,j,c}^{(t)}$ be the predicted velocity at denoising step t, candidate b, dynamic token j, and channel c. For a temporal window of length W, the implementation evaluates

$$\mathrm{CV}_{b,j}^{(t)} = \frac{\frac{1}{C}\sum_c \mathrm{Std}_{k=0}^{W-1}\left[v_{b,j,c}^{(t-k)}\right]}{\frac{1}{WC}\sum_{k=0}^{W-1}\sum_c \left|v_{b,j,c}^{(t-k)}\right| + 10^{-8}}. \tag{12}$$

The standard deviation follows the default sample-standard-deviation convention of the implementation. The reported TVC result uses $W = 3$, a five-step warm-up, and $\tau_{\min} = 0.5$ and $\tau_{\max} = 2.0$. A token is frozen when $\mathrm{CV}_{b,j}^{(t)} < \tau_t$. Freezing is irreversible: a frozen token is removed from subsequent dynamic-token transformer computation and reuses its most recent velocity prediction for all remaining steps. There is no later reactivation or periodic full update in the reported TVC variant. When candidates are processed in a batch, the group-union rule described below applies.

Complexity Discussion Let T be the number of denoising steps, L_{text} the number of text tokens, $L_{\mathcal{G}}$ the number of reusable background tokens, and $L_{\mathcal{D}}$ the number of dynamic tokens, where typically $L_{\mathcal{D}} \ll L_{\mathcal{G}} + L_{\mathcal{D}}$ in image inpainting. Vanilla multi-candidate inference recomputes all image tokens for all N candidates, leading to a total attention cost proportional to

$$\mathcal{O}\big(NT(L_{\mathcal{G}} + L_{\mathcal{D}}) \cdot \big(L_{\text{text}} + L_{\mathcal{G}} + L_{\mathcal{D}}\big)\big). \tag{13}$$

Table 7: **Paired mask-exterior fidelity on 240 EditBench examples.** Intervals are paired percentile-bootstrap 95% confidence intervals with 10,000 resamples. Far-BG and Boundary denote the regions defined above.

Metric	BoN	Ours	Δ (Ours−BoN)	95% CI
Exterior PSNR ↑	28.714	28.157	−0.557	[−0.660, −0.458]
Exterior MAE ↓	0.02919	0.03143	0.00224	[0.00182, 0.00267]
Far-BG MAE ↓	0.02894	0.03097	0.00203	[0.00160, 0.00248]
Boundary MAE ↓	0.03743	0.04459	0.00716	[0.00638, 0.00795]

Our framework performs one full anchor pass and then updates only active dynamic tokens for later candidates. We decompose its cost into the one-time anchor pass and the cost of the remaining candidates:

$$C_{\text{anchor}} = \mathcal{O}\big(T(L_{\mathcal{G}} + L_{\mathcal{D}}) \cdot \big(L_{\text{text}} + L_{\mathcal{G}} + L_{\mathcal{D}}\big)\big)\,, \tag{14a}$$

$$C_{\text{later}} = \mathcal{O}\left(\sum_{n=2}^{N}\sum_{t=1}^{T} |\mathcal{A}_t^{(n)}| \cdot \big(L_{\text{text}} + L_{\mathcal{G}} + |\mathcal{A}_t^{(n)}|\big)\right). \tag{14b}$$

Accordingly, the total cost of our framework is

$$C_{\text{ours}} = C_{\text{anchor}} + C_{\text{later}}. \tag{15}$$

where $|\mathcal{A}_t^{(n)}| \le L_{\mathcal{D}}$ typically decreases over time. The first term corresponds to constructing the anchor-cached context once, while the second term captures the later candidates with both background reuse and temporal active-set shrinking. Therefore, the proposed framework reduces *cross-candidate spatial redundancy* and *within-candidate temporal redundancy* in a unified manner, leading to substantially more efficient inference-time scaling for image inpainting.

D Mask-Exterior Fidelity

Evaluation protocol. We evaluate mask-exterior fidelity on 240 EditBench examples using the same outputs selected under the total trajectory budget of $N = 32$. For both BoN and our method, the final output is selected using the equal-weight Z-score composite reward. Each selected output is compared with its corresponding input image over the observed, unmasked region.

We report four complementary measurements. Exterior PSNR and exterior MAE are computed over all unmasked pixels. Far-background MAE excludes an additional eight-pixel square dilation of the original mask, thereby isolating observed pixels sufficiently far from the editing interface. Exterior-boundary MAE is computed over the four-pixel-wide unmasked ring immediately outside the original mask. Pixel values are normalized to $[0, 1]$.

These measurements characterize pixel-level preservation of the observed image context. In particular, the far-background and exterior-boundary regions allow us to distinguish widespread background drift from localized variation near the editing interface. They do not measure the perceptual naturalness, semantic plausibility, or visual smoothness of the generated content across the mask boundary.

For each metric, we compute the difference between our method and BoN on each of the 240 paired examples. The reported uncertainty intervals are 95% paired percentile-bootstrap confidence intervals obtained with 10,000 resamples.

Background preservation and boundary-localized adaptation. A central component of our spatial reuse design is the explicit dilation of the target mask. As described in the main paper, tokens inside the target region and the two-token boundary band are retained in the dynamic set and recomputed for each candidate. Tokens sufficiently far from the editing interface instead reuse the anchor-cached background states.

This separation is designed to satisfy two requirements of image inpainting. First, the observed background should remain stable and continue to provide fixed visual context for the generated region. Second, tokens close to the mask boundary should not be prematurely frozen, because they mediate the local interaction between the newly generated content and its surrounding image.

Table 8: **Metrics used for candidate selection and evaluation.** The composite selector equally averages the normalized CLIP Score, HPSv3, and VQA Score. The weights are fixed across all methods, samples, and candidate budgets.

Metric	Role	Weight w_m
CLIP Score	Candidate selection	$1/3$
HPSv3	Candidate selection	$1/3$
VQA Score	Candidate selection	$1/3$
HPS2.1	Evaluation only	–
PickScore	Evaluation only	–
ImageReward	Evaluation only	–

The paired fidelity results exhibit this intended spatial pattern. Far-background MAE changes from 0.02894 for BON to 0.03097 for our method, corresponding to a difference of 0.00203. In comparison, the difference within the narrow exterior-boundary ring is 0.00716. The boundary-localized difference is therefore substantially larger than the difference measured over the distant background.

This result indicates that our method does not induce a uniform or widespread shift over the observed image context. Instead, the measurable variation is concentrated near the editing interface, where local features must remain responsive to the generated content. The distant background remains considerably more stable, consistent with its intended role as reusable observed context.

Although the image-space diagnostic rings used in this evaluation are not identical to the two-token boundary band used inside the FLUX transformer, they provide a complementary spatial measurement of the same design principle: adaptation should remain localized around the editing interface, while the remaining background should be preserved and reused.

Over the complete unmasked region, exterior PSNR changes from 28.714 dB to 28.157 dB, while exterior MAE changes from 0.02919 to 0.03143. When interpreted together with the far-background and boundary-specific measurements, these aggregate values show that the observed differences are not distributed uniformly throughout the unmasked image. Instead, the far-background branch remains substantially more stable than the immediate mask exterior.

Importantly, the purpose of the dynamic boundary band is not to modify the known background arbitrarily. Its purpose is to prevent the context closest to the editing interface from being incorrectly treated as fully independent of the generated region. By retaining this narrow boundary neighborhood in the dynamic set, the method preserves the global role of the unmasked background as observed context while allowing the local interaction required for coherent inpainting. The resulting spatial behavior is therefore consistent with the fundamental requirement of image inpainting: preserve the known background globally while adapting only the local interface needed to integrate the newly generated content.

E Composite Reward Construction

Selection metrics. For all methods, the final output is selected using a fixed composite reward constructed from three evaluators: CLIP Score, HPSv3, and VQA Score. These metrics provide complementary signals for text–image alignment, human preference, and instruction satisfaction. The remaining evaluation metrics, namely HPS2.1, PickScore, and ImageReward, do not participate in candidate selection and are used only for reporting the quality of the selected output.

Let $q_{s,i,m}$ denote the full-image score assigned to candidate i of sample s by selector metric

$$m \in \mathcal{M}_{\text{sel}} = \{\text{CLIP}, \text{HPSv3}, \text{VQA}\}.$$

All selector rewards are computed on the full generated image. We do not use mask-region crops or metric-specific local rewards in the composite selector.

Candidate alignment and normalization scope. Candidates from the three selector-metric manifests are aligned by candidate index and, when available, random seed. A candidate is included in the selection pool only when all three selector scores are available.

Normalization is performed independently for each input sample, method, selector metric, and reported candidate budget. For clarity, the method and candidate-budget indices are omitted from the notation below. Let $\mathcal{C}_s$ denote the candidate pool available to the selector for sample s. It contains N independently generated trajectories for Global BoN, or one anchor trajectory and $N-1$ reuse trajectories for an anchor-based method. Candidates generated by other methods, candidates from other input samples, and candidates outside the reported budget are never included in the normalization statistics.

This normalization scope ensures that the selected output is independent of which comparison methods are included in a quantitative table. It also prevents results obtained from a larger candidate pool from affecting selection under a smaller reported budget.

For each selector metric, we compute the population mean and standard deviation over the available candidate pool:

$$\mu_{s,m} = \frac{1}{|\mathcal{C}_s|} \sum_{j \in \mathcal{C}_s} q_{s,j,m}, \tag{16}$$

$$\sigma_{s,m} = \sqrt{\frac{1}{|\mathcal{C}_s|} \sum_{j \in \mathcal{C}_s} (q_{s,j,m} - \mu_{s,m})^2}. \tag{17}$$

The corresponding normalized score is

$$z_{s,i,m} = \frac{q_{s,i,m} - \mu_{s,m}}{\sigma_{s,m}}, \qquad \sigma_{s,m} \geq \epsilon, \qquad \epsilon = 10^{-8}. \tag{18}$$

If $\sigma_{s,m} < \epsilon$, the normalized scores of that metric are set to zero for all candidates in the pool, since the metric does not distinguish among the available candidates.

For the marked early-pruning simulations, the normalization statistics are computed once from the initial pool of 32 Global trajectories and remain fixed throughout the pruning process. Final selection is restricted to the surviving trajectories, without post-hoc renormalization.

Equal-weight Z-score selector. The composite reward is defined as the equal-weight average of the three normalized selector metrics:

$$S_{s,i} = \frac{1}{3} \left(z_{s,i,\mathrm{CLIP}} + z_{s,i,\mathrm{HPSv3}} + z_{s,i,\mathrm{VQA}} \right). \tag{19}$$

The selected candidate is

$$i_s^* = \arg\max_{i \in \mathcal{C}_s} S_{s,i}. \tag{20}$$

The same selector and equal weights are used for every evaluated method and candidate budget. The weights are fixed and are not fitted on EditBench or BrushBench. Therefore, $S_{s,i}$ should be interpreted as a relative within-sample ranking score rather than an absolute measure of image quality.

After candidate selection, all six full-image evaluation metrics are computed or retrieved for the same selected candidate i_s^*. We do not independently select a different oracle candidate for each evaluation metric.

F User Study

Study objective. We conduct a blinded pairwise user study to complement the automatic evaluation and assess whether human observers prefer the results generated by our method. The study focuses on two aspects central to text-guided image inpainting: (i) whether the generated content correctly follows the inpainting instruction, and (ii) whether the completed image is visually plausible and aesthetically consistent with its surrounding context.

Unlike automatic evaluation, participants are not asked to assign absolute numerical scores. Each trial compares the output of our method with that of one baseline method under the same input image, mask, and text instruction. Participants select the result that is better overall or indicate that there is no clear preference between the two. This protocol directly produces the *Ours–Tie–Baseline* preference percentages reported in Tab. 9.

Participants. The study is administered through a questionnaire to participants who did not take part in this research. A total of 30 participants completed the questionnaire. The identities of the evaluated methods are not disclosed to the participants, and no member of the research team contributes ratings to the reported results.

Each complete questionnaire contains 50 pairwise-comparison trials, consisting of five source examples for each of the ten baseline methods evaluated in the main paper. With 30 participants completing all trials, the study contains 1,500 individual judgments in total.

Stimulus construction. We select 50 text-guided image inpainting examples from the combined evaluation pools of EditBench and BrushBench. The examples are sampled uniformly using a fixed random seed. Each example consists of a masked input image, its corresponding inpainting instruction, and the generated outputs to be compared.

The 50 examples are divided among the ten baseline comparisons, with five source examples assigned to each method pair. For a given baseline, the same five source examples are presented to all participants, ensuring that all participants evaluate the same set of pairwise stimuli for that comparison.

For our method, we use the same $N = 32$ output selected by the equal-weight Z-score composite reward. Specifically, the selector averages the normalized CLIP Score, HPSv3, and VQA Score. For each baseline, we use the corresponding output produced under the evaluation setting reported in the main paper. No result is manually selected, replaced, or modified for the user study.

Each trial contains exactly two generated images: one from our method and one from a single baseline. Participants are therefore not required to rank all methods simultaneously. Comparisons with different baselines are administered as separate pairwise trials.

Randomization and blinding. For each trial, the two generated results are displayed side by side together with the masked input image and its inpainting instruction. Model names, automatic evaluation scores, inference costs, and other identifying information are not shown.

The left–right placement of our result and the baseline result is independently randomized with equal probability for every trial. The presentation order of the 50 trials within each questionnaire is also randomized. Consequently, neither a fixed display position nor the questionnaire order systematically favors a particular method.

The randomization assignment is recorded so that the anonymous left- and right-side responses can be mapped back to the corresponding methods during result aggregation.

Evaluation instructions. For each trial, participants are asked the following question:

> *Given the masked input image and the inpainting instruction, which of the two generated results is better overall?*

Participants are instructed to jointly consider:

1. whether the generated content correctly satisfies the text instruction; and
2. whether the result contains visible artifacts, distorted structures, or unnatural transitions near the mask boundary.

The available responses are:

Left is better, *Tie*, *Right is better*.

The tie option is retained to avoid forcing an artificial preference when the two results have comparable quality. Participants jointly consider instruction following and visual quality and submit one overall preference for each trial, rather than providing separate scores for the two criteria.

Questionnaire assignment. Each participant evaluates all 50 pairwise-comparison trials, covering the five source examples associated with each of the ten baseline comparisons. With 30 participants, this results in 1,500 individual judgments across the complete user study.

Table 9: **Pairwise user-study results.** For each baseline, participants compare its output with the output of our method and choose whether our result is better, the two results are tied, or the baseline result is better.

Baseline	Ours (%)	Tie (%)	Baseline (%)
SD2I	87.4	3.1	9.5
CNI	85.0	7.5	7.5
BLD	83.1	9.4	7.5
FreePaint	87.5	3.8	8.8
PowerPaint	88.0	3.9	8.1
PrefPaint	85.6	4.4	10.0
HD-Painter	81.9	5.6	12.5
ASUKA	69.4	15.6	15.0
BrushNet	76.9	7.5	15.6
FLUX.1 Fill	75.0	5.6	19.4

The assignment is balanced so that every baseline receives the same number of evaluation opportunities. The randomized presentation procedure also ensures that our method and the corresponding baseline appear approximately equally often on the left and right sides.

Each participant evaluates a given unordered image pair only once. The order of all method pairs and source examples is independently randomized within each questionnaire.

Preference aggregation. For baseline method b, let W_b, T_b, and L_b denote the numbers of valid judgments preferring our method, reporting a tie, and preferring the baseline, respectively. The total number of judgments for the comparison is

$$J_b = W_b + T_b + L_b. \tag{21}$$

The three percentages reported in Tab. 9 are computed as

$$P_b^{\text{ours}} = 100\frac{W_b}{J_b}, \qquad P_b^{\text{tie}} = 100\frac{T_b}{J_b}, \qquad P_b^{\text{base}} = 100\frac{L_b}{J_b}. \tag{22}$$

Ties are reported as a separate category and are not divided between the two methods. Each valid response contributes exactly once to the comparison with the corresponding baseline.

User-study results. Tab. 9 summarizes the pairwise preference percentages. Our method receives the largest share of preferences in every comparison. Depending on the baseline, the preference rate for our method ranges from 69.4% to 88.0%, whereas the baseline preference rate ranges from 7.5% to 19.4%. The proportion of ties ranges from 3.1% to 15.6%.