SDE TRAJECTORY STEERING WITH SPATIAL-AWARE VLM SELF-VERIFICATION: TEST-TIME SCALING FOR IMAGE EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time inference strategies for diffusion-based editing currently face distinct bottlenecks. Standard ODE sampling is rigid and cannot correct early errors; Best-of-N scaling is computationally prohibitive; and recent tree-search alternatives introduce external reward semantic discrepancies, devolving into undirected stochastic sampling via directionless *prune-and-random-restart* mechanisms. To address these compounding limitations, we propose SDESTEER, a novel training-free test-time scaling framework. Our approach replaces blind noise injection with an adaptive *prune-and-steer* mechanism. By progressively pruning misaligned sampling paths and dynamically modulating the classifier-free guidance (CFG) scale based on an intermediate verification heuristic, SDESTEER actively steers SDE-induced noise to synthesize novel, optimized sampling trajectories. Furthermore, we achieve a zero-overhead architecture by dual-purposing the native Vision-Language Model (VLM) backbone as both the text encoder and an internal verifier. To ensure reliable evaluation without auxiliary dependencies, we introduce a spatially-aware verification strategy that effectively mitigates the self-enhancement bias typical of same-model assessment. Extensive experiments demonstrate that our actively guided test-time compute translates directly into superior generative alignment and visual quality.

1 INTRODUCTION

Many recent instruction-guided image editing methods based on Diffusion models Ho et al. (2020); Song et al. (2021); Brooks et al. (2023) typically rely on deterministic ODE sampling. However, these rigid trajectories struggle to recover from early-stage errors or sub-optimal manifolds. Motivated by test-time scaling (TTS) in large language models OpenAI (2024); Snell et al. (2024), recent efforts explore scaling inference compute for diffusion models Ma et al. (2025). The standard baseline, Best-of-N (BoN) Ma et al. (2025); Xie et al. (2025); Shen et al. (2025), mitigates worst-case failures by selecting the best of multiple independent outputs. Yet, BoN is computationally prohibitive and acts merely as a passive post-hoc filter, fundamentally failing to synthesize novel, optimized trajectories.

To address BoN's inefficiency, recent methods adopt tree-structured search Kim et al. (2025c); Jain et al. (2025); Guo et al. (2025); He et al. (2025), pruning unpromising branches at intermediate checkpoints to concentrate computation. While improving the success-to-computation ratio, they introduce two critical bottlenecks. First, relying on external reward models Wu et al. (2025b) introduces a severe "knowledge discrepancy": disparate learned priors between the editing and verification models cause semantic misalignment. Second, current frameworks lack directional guidance. By duplicating surviving branches and injecting random noise (e.g., standard SDE resampling), this *prune-and-random-restart* paradigm devolves into undirected stochastic sampling, struggling to actively correct misaligned features.

To address these compounding limitations, we propose SDESTEER, a novel training-free test-time scaling framework that transforms passive generation into active, guided exploration. SDESTEER replaces rigid ODE paths and blind SDE guessing with an adaptive *prune-and-steer* mechanism. Guided by an intermediate verification heuristic, our framework prunes misaligned branches and

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.

dynamically modulates the Classifier-Free Guidance (CFG) scale based on the verification score. This modulation actively steers SDE-induced noise to synthesize novel, high-fidelity sampling paths unreachable via standard inference. Furthermore, to resolve the knowledge discrepancy, we design a unified self-verification architecture. We dual-purpose the native Vision-Language Model (VLM) backbone—already integrated within modern diffusion editors—as both the multimodal encoder and the internal verifier. Recognizing that same-model evaluation inherently risks self-enhancement bias, we introduce a spatially-aware verification strategy. This effectively grounds the self-evaluation in spatial realities, eliminating auxiliary reward dependencies and ensuring robust, objective pruning.

The main contributions are summarized as follows:

- We propose SDESTEER, a training-free test-time scaling framework that overcomes the rigidity of ODE sampling, the computational inefficiency of Best-of-N, and the undirected nature of existing tree-search paradigms.
- We introduce an adaptive *prune-and-steer* mechanism. By progressively pruning misaligned branches and dynamically modulating the CFG scale based on verification scores, we actively steer SDE-induced noise to synthesize novel, high-fidelity trajectories.
- We design a unified self-verification architecture by dual-purposing the native VLM backbone as both the multimodal encoder and internal verifier. Coupled with a spatially-aware verification strategy, we effectively mitigate self-enhancement bias and eliminate auxiliary dependencies.

2 RELATED WORK

Test-Time Scaling and Tree Search for Diffusion. Test-time scaling aims to improve generation quality by allocating additional computation across candidate trajectories during sampling (Ma et al., 2025). Early approaches like Best-of-N suffer from severe computational redundancy. To concentrate computation on promising candidates, recent works have explored reward-guided diffusion, beam search, and evolutionary search to branch or mutate trajectories (Yeh et al., 2025; Singhal et al., 2025; Zhang et al., 2025a; Jain et al., 2025; Oshima et al., 2025; He et al., 2025). Furthermore, methods like FlowITS (Kim et al., 2025a) attempt to restore stochastic exploration within deterministic flows. However, when these tree-based methods prune unpromising branches, they typically rely on random noise injection for subsequent exploration. This "prune-and-random-restart" paradigm acts as undirected blind guessing. In contrast, SDESTEER employs *shared-prior SDE branching* to anchor structural exploration, and translates intermediate verification scores into dynamic CFG modulation. By actively warping the reverse SDE drift term, our method explicitly *steers* trajectories along the semantic gradient rather than randomly guessing, synthesizing highly optimized editing paths.

Self-Verification and Active Steering. While instruction-guided editing has evolved rapidly via diffusion and multimodal flow models (Brooks et al., 2023; Wu et al., 2025a), ensuring test-time reliability remains challenging. Methods like ICEdit, ELECT, and PPAD (Zhang et al., 2025b; Kim et al., 2025b; Lv et al., 2025) screen independent candidates or formulate corrections using external MLLMs. However, these external paradigms act as passive post-hoc filters and introduce severe knowledge discrepancy. To avoid external architectural overhead, works like dMLLM-TTS (Xin et al., 2026) utilizes a model's internal capabilities for self-verification, though this notoriously suffers from self-enhancement bias and proxy overoptimization (Ku et al., 2024). SDESTEER bridges these gaps by dual-purposing the native VLM as a unified adjudicator to eliminate knowledge discrepancy. To explicitly combat self-enhancement bias, we delay pure semantic VLM evaluation until mid-stage previews become interpretable, relying instead on direct structural latent comparisons during critical early stages. Furthermore, our mid-stage verification is anchored by a *spatially-aware mask*, grounding the VLM in structural reality. By coupling this robust self-evaluation with our adaptive *prune-and-steer* mechanism, our framework fundamentally shifts the paradigm from passive filtering to active, direction-aware trajectory synthesis.

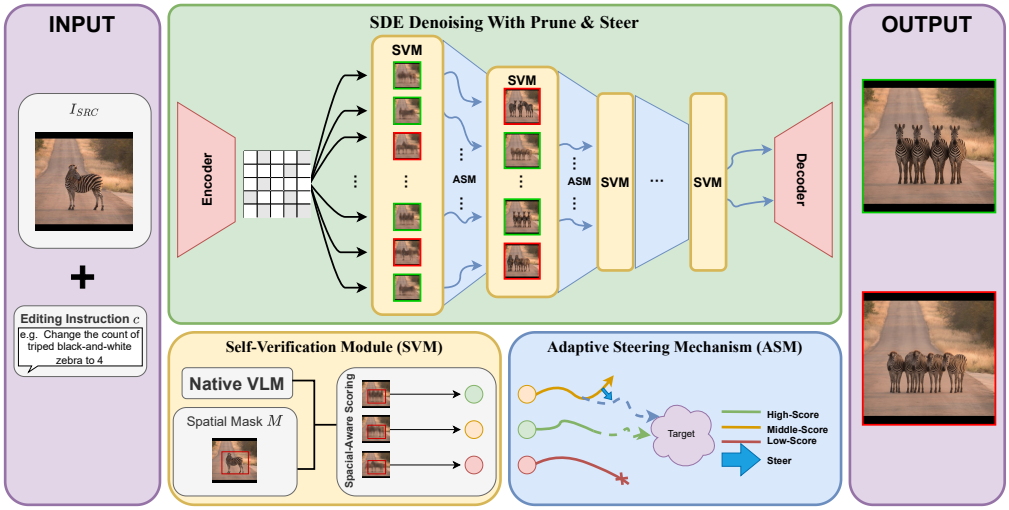

Figure 1: **Overview of the SDESTEER pipeline.** By casting diffusion editing as the exploratory steering of SDE trajectories, our method utilizes VLM self-verification and active steering to continuously align surviving branches with the editing instruction.

3 METHODOLOGY

3.1 PROBLEM FORMULATION AND OVERVIEW

Given a source image I_{src} and an editing instruction c, our goal is to generate an edited image that strictly adheres to the instruction while preserving irrelevant source regions. As discussed previously, standard deterministic sampling and existing test-time scaling methods (e.g., Best-of-N or undirected tree search) struggle with trajectory rigidity, semantic knowledge discrepancies, and passive exploration.

To overcome these limitations, we introduce SDESTEER, a novel training-free test-time scaling architecture that transforms passive sampling into an actively guided exploration. As illustrated in Figure 1, in the initial step, the VLM processes I_{src} and c to simultaneously extract cross-modal semantic features and a spatial mask M, which precisely delineates the instruction-relevant regions. Guided by these extracted features, we anchor the diffusion process to a shared structural prior, denoted as x_T, and spawn multiple Stochastic Differential Equation (SDE) branches to explore local variations. During this traversal, we perform stage-aware self-verification at discrete checkpoints. Crucially, we dual-purpose the native Vision-Language Model (VLM) backbone as both the multimodal encoder and the internal verifier; this unified design entirely eliminates the semantic knowledge discrepancies introduced by external reward models. Ultimately, the verification scores derived across these stages establish an active *prune-and-steer* mechanism. By dynamically modulating the Classifier-Free Guidance (CFG) scale based on these scores, SDESTEER actively steers the surviving SDE paths toward the target text manifold.

3.2 SHARED-PRIOR SDE BRANCHING

To ensure that our test-time search space evaluates genuine editing progress rather than arbitrary variations, we anchor the origin of our search tree to a shared structural prior, x_T. From this unified topological root, we initiate the diffusion process and spawn K independent Stochastic Differential Equation (SDE) branches. For a branch k at timestep t, the update incorporates an independent Wiener process $w^{(k)}$:

$$dx^{(k)} = f(x^{(k)}, t, c, I_{\text{src}})dt + g(t)dw^{(k)}. \tag{1}$$

This shared-prior design is critical for reliable stage-aware verification. Conventional *prune-and-random-restart* paradigms typically spawn trajectories from independent noise initializations. This fundamentally confounds initialization variance with true generative capability—a branch may

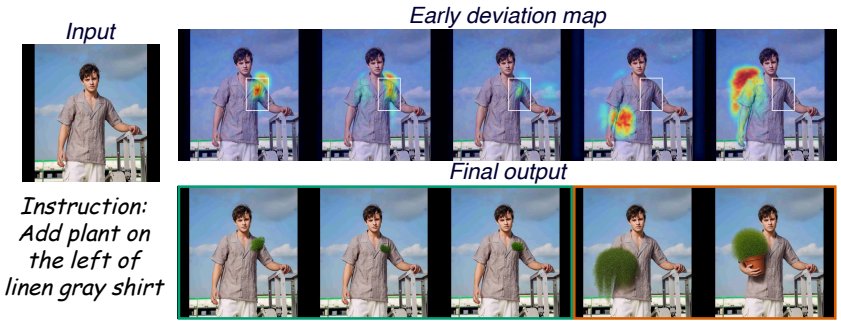

Figure 2: Early-stage spatial scoring. Green and orange borders denote high- and low-scoring branches respectively.

survive merely due to a fortuitous initial noise draw rather than meaningful semantic alignment. By anchoring all SDE branches to a single x_T, SDESTEER strictly isolates localized semantic evolution from structural randomness. This guarantees that the variance across branches stems solely from the stochastic sampling paths, transforming the exploration process from a passive noise lottery into a rigorous objective evaluation.

3.3 SPATIAL-AWARE VLM SELF-VERIFICATION

A major bottleneck in existing tree search methods is the reliance on external reward or segmentation models for intermediate scoring, which introduces semantic knowledge discrepancy and significant deployment overhead. To eliminate these issues, we design a unified self-verification architecture, exploiting the generative model's native VLM to ground the verification process with no auxiliary-model overhead.

Before traversing the SDE branches, the source image I_{src} and the instruction c are fed into the native VLM. While standard pipelines utilize this step solely to extract cross-modal semantic features, we concurrently extract a spatial attention mask $M \in \{0,1\}^{H\times W}$. This mask naturally delineates the regions targeted for editing ($M = 1$) from the background meant for preservation ($M = 0$). By extracting this spatial prior during the standard conditioning phase, we establish a robust foundation for our subsequent self-verification, entirely eliminating the need for auxiliary vision models.

To optimize test-time scaling within a constrained inference budget, we employ an iterative prune-and-steer strategy across the denoising process. We initiate the SDE tree with $K = 8$ parallel sampling paths originating from the shared prior x_T. At the early stage (τ_{early}), we evaluate the initial trajectories, prune the 2 lowest-scoring branches to retain 6. During the middle stage (τ_{mid}), we leverage a spatial-aware VLM self-verifier to iteratively score, steer, and prune the candidate pool—discarding 2 paths per evaluation step—until only 2 paths remain. These 2 surviving trajectories are denoised uninterrupted to the final step ($t = 0$) to obtain the x_0 latents, which are then decoded into image space. In the final stage, the spatial-aware VLM evaluates the decoded images to select the optimal final output.

Early-Stage Spatial Scoring. During the early stage of reverse sampling ($t = \tau_{\text{early}}$), the predicted clean latent $\hat{x}_0$ remains semantically ambiguous and is therefore unsuitable for reliable VLM-based evaluation. Nevertheless, its spatial deviation from the source latent already provides a useful structural signal. We thus define a purely spatial score $\mathcal{S}^{(k)}_{\text{early}}$ by measuring the latent deviation between the predicted clean latent $\hat{x}^{(k)}_0$ of branch k and the source latent x_{src}, using only the prior mask M.

Let $U(\cdot)$ restore packed latent tokens to a spatial grid. For each spatial location p, we extract

$$u^{(k)}(p) = U(\hat{x}^{(k)}_0)(p), \qquad u_{\text{src}}(p) = U(x_{\text{src}})(p). \tag{2}$$

We then compute the normalized ℓ_2 deviation at each location:

$$d_{\mathrm{nL2}}^{(k)}(p) = \frac{\left\|u^{(k)}(p) - u_{\mathrm{src}}(p)\right\|_2}{\max\left\{\|u_{\mathrm{src}}(p)\|_2, \varepsilon\right\}}, \tag{3}$$

where ε is a small constant for numerical stability. Using the prior mask M, we measure target activation inside the edit region and unintended change outside it:

$$A^{(k)} = Q_{0.9}\left(\left\{d_{\mathrm{nL2}}^{(k)}(p) \mid M(p) = 1\right\}\right), \tag{4}$$

$$C^{(k)} = \frac{1}{|\overline{M}|} \sum_{p:M(p)=0} d_{\mathrm{nL2}}^{(k)}(p), \tag{5}$$

where $Q_{0.9}$ denotes the empirical 90th percentile and $\overline{M}$ denotes the spatial region outside the mask. The activation statistic $A^{(k)}$ captures whether sufficiently strong changes emerge within the target region, whereas $C^{(k)}$ measures the average deviation in regions that should remain unchanged. Finally, the early-stage spatial score is then defined as

$$\mathcal{S}_{\mathrm{early}}^{(k)} = A^{(k)} \cdot (1 - C^{(k)}) \tag{6}$$

As Figure 2 illustrates, SDE branches with early latent deviations localized within the VLM-derived mask M reliably yield precise edits, whereas diffuse activations produce misplaced objects. This validates our stage-aware strategy: although early noisy states (x_t) lack semantic clarity, their structural trajectories are highly predictive. Intersecting these latent deviations with M thus provides a robust, sibling-calibrated pruning signal, efficiently filtering unpromising paths and anchoring structural exploration prior to computationally intensive, mid-stage semantic verification.

Mid-Stage and Spatial-Aware VLM Scoring. As the reverse diffusion progresses to the mid-stage (τ_{mid}) and beyond, we project the noisy state to obtain a preview of the clean latent $\hat{x}_0^{(k)}$. Because this preview now attains structural coherence, low-level metrics are insufficient for semantic evaluation. To preclude external knowledge discrepancy and maintain a zero-overhead architecture, we repurpose the native VLM backbone, V_θ, as an internal semantic adjudicator.

To systematically overcome the inherent self-enhancement bias of same-model evaluation, we implement a spatially-aware, multi-dimensional verification mechanism anchored by the *a priori* mask M. We construct a suite of dimension-specific prompts, denoted as $P = \{p_{\mathrm{IF}}, p_{\mathrm{BP}}, p_{\mathrm{IQ}}\}$, which respectively target Instruction Following (IF), Background Preservation (BP), and Image Quality (IQ). These prompts dynamically encapsulate the original editing instruction c into structured assessment criteria. Let $\tilde{I}^{(k)} = \mathrm{Dec}(\hat{x}_0^{(k)})$ denote the decoded preview. Defining $\mathbf{s}^k = (s_{\mathrm{IF}}^k, s_{\mathrm{BP}}^k, s_{\mathrm{IQ}}^k)$, the spatially-aware verification is written as:

$$\mathbf{s}^k = V_\theta\big(I_{\mathrm{src}}, \tilde{I}^{(k)}, M, P\big) \tag{7}$$

where V_θ maps the spatially constrained inputs to continuous scores bounded in $[0, 1]$. Specifically, s_{IF} evaluates semantic alignment strictly within the target region ($M = 1$), s_{BP} quantifies contextual fidelity exclusively within the background ($M = 0$), and s_{IQ} targets perceptual realism.

We combine these dimensions into a unified VLM score in which instruction following and background preservation act as multiplicative bottlenecks, while image quality provides logarithmic modulation:

$$\mathcal{S}_{\mathrm{vlm}}^{(k)} = s_{\mathrm{IF}}^k \cdot s_{\mathrm{BP}}^k \cdot \ln\left(1 + s_{\mathrm{IQ}}^k\right) \tag{8}$$

Ultimately, the score $\mathcal{S}_{\mathrm{vlm}}^{(k)}$ is not merely used for pruning to decide branch survival. It actively governs trajectory adjustments and serves as the ultimate criterion for optimal output selection.

Final-Stage Verification. Once the two surviving latents are decoded into image space, the spatial-aware VLM performs a definitive region-aware evaluation to select the optimal image as the final output.

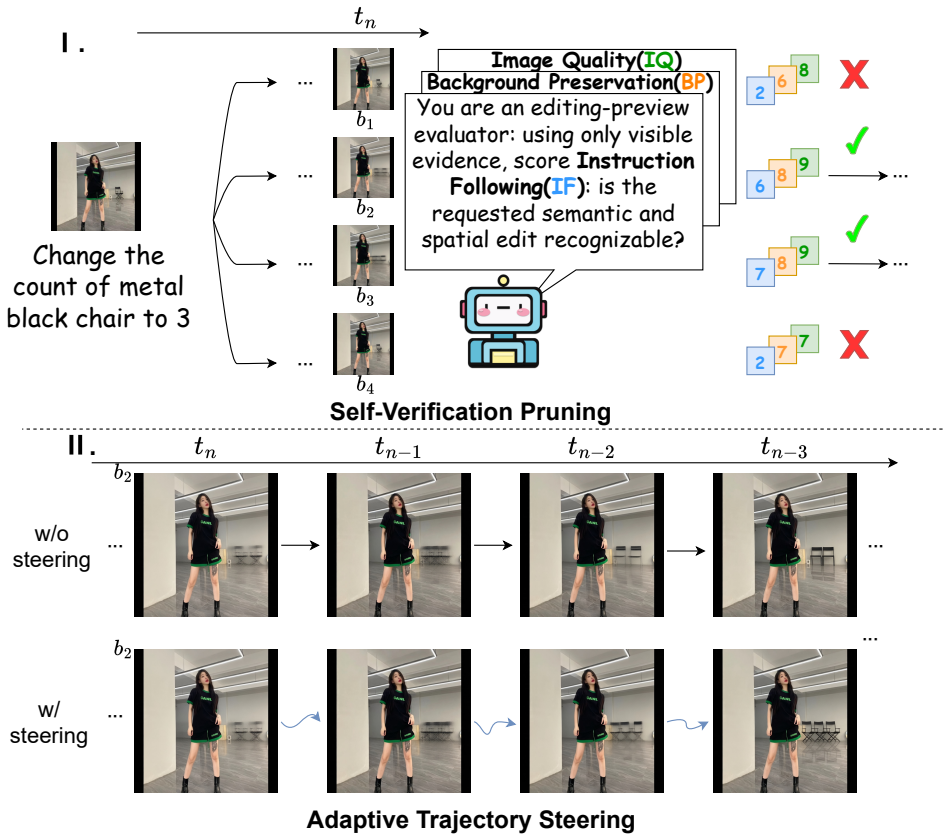

Figure 3: Self-verification based pruning and adaptive trajectory steering.

3.4 Adaptive Prune-and-Steer via CFG Modulation

Existing test-time scaling methods typically execute a passive "prune-and-random-restart" strategy: they discard low-scoring branches and duplicate survivors by injecting isotropic Gaussian noise. Because this noise is undirected, the duplicated branches merely perform a blind random walk. This inherently inefficient strategy relies on chance to stumble out of poor local minima, failing to synthesize genuinely novel semantic structures.

In contrast, SDESteer closes the loop by introducing an active *prune-and-steer* mechanism (illustrated in Figure 3). After verification, the unified VLM score $\mathcal{S}_{\text{vlm}}^{(k)}$ dictates overall branch survival. For the surviving branches, we dynamically modulate the Classifier-Free Guidance (CFG) scale based on their specific instruction following (IF) and background preservation (BP) scores. The branch-specific steering magnitude $\eta^{(k)}$ and the modulated CFG scale $\omega^{(k)}$ are formulated as:

$$\eta^{(k)} = (1 - s_{\text{IF}}^{k}) \cdot s_{\text{BP}}^{k}, \tag{9}$$

$$\omega^{(k)} = \omega_{\text{base}} + \gamma \eta^{(k)} \tag{10}$$

where ω_{base} is the standard guidance scale and γ is a hyperparameter governing the maximum steering capacity.

This dynamically modulated scale $\omega^{(k)}$ is applied directly to the noise prediction:

$$\tilde{\epsilon}_{\theta}^{(k)} = \epsilon_{\theta}(x_t^{(k)}, t, \emptyset) + \omega^{(k)} \underbrace{\left[\epsilon_{\theta}(x_t^{(k)}, t, c) - \epsilon_{\theta}(x_t^{(k)}, t, \emptyset)\right]}_{\text{Semantic Gradient}} \tag{11}$$

Direction-Aware Novel Path Synthesis. Unlike standard tree search that relies solely on the undirected Wiener process $g(t)dw^{(k)}$, our approach actively warps the reverse SDE *drift* term. The

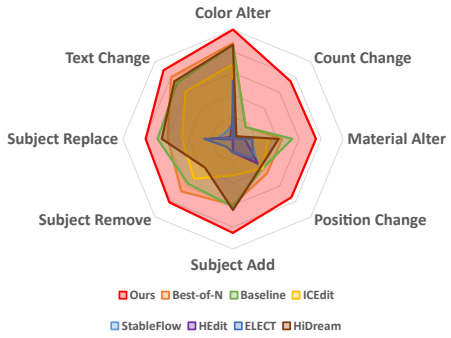

Figure 4: Success rates across 8 editing tasks. Larger radius denotes higher success.

Table 1: Quantitative evaluation results.

Method	IF↑	BP↑	IQ↑	CLIP-I↑	CLIP-T↑	SR(%)↑
ICEdit	8.25	9.44	9.75	0.91	0.20	41.3
HiDream-E1	8.34	9.57	**9.77**	0.90	0.20	50.8
StableFlow	6.42	9.45	9.71	0.93	0.19	12.5
h-Edit	6.65	8.20	8.51	0.66	0.20	12.6
ELECT	7.54	9.46	9.73	0.86	**0.21**	18.4
Baseline	8.18	9.56	9.74	0.94	0.19	56.5
Best-of-N	8.20	9.57	9.76	**0.96**	0.19	57.2
SDESTEER	**8.55**	**9.62**	**9.77**	0.94	0.19	**82.4**

Table 2: Human evaluation results.

Method	IF↑	BP↑	IQ↑	SR(%)↑
ICEdit	5.70	6.70	7.57	27.6
HiDream-E1	6.37	7.32	7.89	37.4
Baseline	7.36	8.20	8.40	53.2
Best-of-N	7.93	8.38	8.56	64.4
SDESTEER	**8.91**	**9.15**	**9.19**	**87.8**

update vector mathematically approximates the implicit classifier gradient, $\nabla_{x_t} \log p(c|x_t)$, steering the generative trajectory directly toward the target text manifold. Critically, this formulation acts as an *adaptive error-correction* mechanism. When a branch preserves the background but lacks target semantics ($s^k_{\text{BP}} \to 1, s^k_{\text{IF}} \to 0$), it aggressively amplifies the steering magnitude $\eta^{(k)}$ to strictly guide the trajectory along the semantic gradient; *vice versa* ($s^k_{\text{IF}} \to 1$ or $s^k_{\text{BP}} \to 0$), $\eta^{(k)}$ gracefully decays to zero, effectively shielding against CFG-induced saturation and topological collapse. As illustrated in Figure 3, despite sharing nearly identical starting states, methods without steering consistently fail to alter the chair count as described. In contrast, our steering strategy adaptively adjusts the subsequent diffusion trajectory to ensure accurate editing.

By coupling spatially-aware verification scores with this implicit gradient, SDESTEER transforms test-time scaling from a passive filter into an active exploration framework. Instead of waiting for undirected noise to synthesize missing features, paths are proportionately driven toward condition c, successfully escaping deterministic ODE constraints to ensure strict semantic adherence and optimal image fidelity.

4 EXPERIMENTS AND EVALUATION

Benchmark and Methods. We use the source-image and instruction pairs from *EdiVal-Bench* (Chen et al., 2025), a human-annotated benchmark covering diverse instruction-guided image editing tasks. Its source–instruction formulation directly aligns with our evaluation setting. We implement SDESTEER using Qwen-Image-Edit-2511 (Wu et al., 2025a) as our base model. We compare our approach against a deterministic ODE baseline (representing zero test-time scaling) and a standard Best-of-N strategy, which blindly explores N random sampling trajectories ($N = 8$ by default) and selects the best final result. In addition, we evaluate ICEdit, HiDream-E1, StableFlow, h-Edit, and ELECT using their official implementations and recommended inference settings.

Implementation details. SDESTEER uses a 50-step schedule with eight branches initialized from a shared Gaussian root. The branches diverge through independent SDE innovations generated using fixed branch-level seeds 141–148 and are progressively pruned $8 \to 6 \to 4 \to 2$ at steps 15, 25, and 35. At each checkpoint, pruning is performed first, after which active, source-anchored trajectory steering is applied to the surviving branches. All experiments were conducted on a single NVIDIA RTX PRO 6000 GPU with 96 GB of memory.

Quantitative Results. To reduce evaluator coupling and mitigate self-preference bias, we employ `gemini-3.1-pro` exclusively as an external evaluator. Its advanced multimodal reasoning pro-

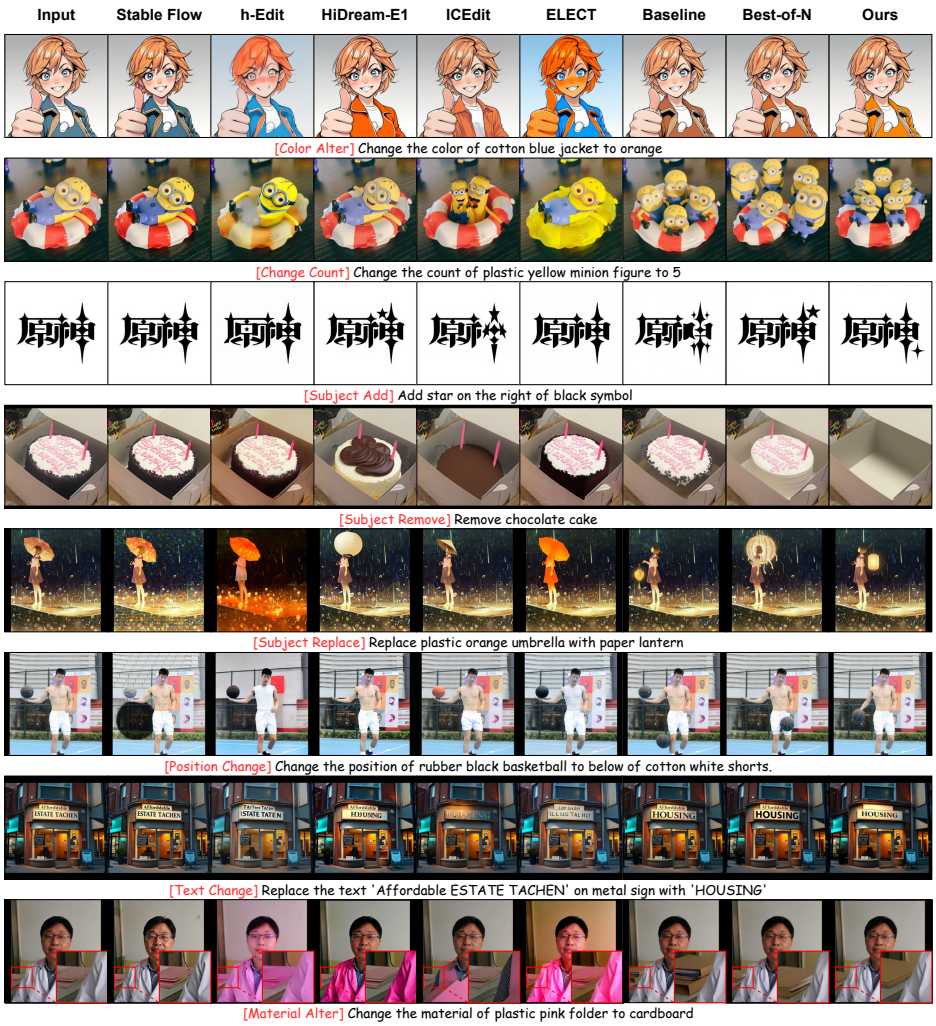

Figure 5: Qualitative comparison across state-of-the-art instruction-guided image editing methods on eight representative tasks.

vides a fair, decoupled benchmark. Given a source image, target instruction, and final edited output, the external VLM assigns $[1, 10]$ scores for instruction following (IF), background preservation (BP), and image quality (IQ), while predicting a binary success label with textual justification. We additionally report CLIP-I for visual preservation and CLIP-T for text alignment as auxiliary metrics, although they may overlook fine-grained editing constraints. We report dataset-level means, with SR computed as the percentage of positive success labels.

Table 1 reports the comprehensive quantitative results of our approach against state-of-the-art baselines. Under the matched Qwen-Image-Edit-2511 backbone, SDESTEER achieves an 82.4% Success Rate (SR), exceeding Best-of-N (57.2%) by 25.2 percentage points. Furthermore, SDESTEER attains the highest Instruction Following (8.55) and Background Preservation (9.62) scores, while tying HiDream-E1 for the highest Image Quality (9.77). Beyond these aggregate metrics, Figure 4 presents a category-wise performance breakdown. This analysis indicates that SDESTEER's improvements occur across multiple editing categories rather than being driven by a single task type.

Human Evaluation. We conducted a user study with 40 participants aged 23–50, comparing five editing methods across eight tasks. Participants viewed randomized, anonymized outputs with the source image and instruction, rated Instruction Following, Background Preservation, and Image Quality, and provided an overall binary Success/Failure judgment. As Table 2 shows, SDESTEER

Table 3: Ablation study of our core components.

Variant	IF↑	BP↑	IQ↑	SR(%)↑
SDESTEER	**8.55**	**9.62**	**9.77**	**82.4**
w/o steering	8.22	9.20	9.75	77.6
w/o spatial awareness	7.67	8.22	9.66	74.4
w/o native self-verification	8.09	9.05	9.22	61.1

achieved an 87.8% success rate, compared with 64.4% for the strongest baseline. Participants also rated instruction following, background preservation, and image quality.

Qualitative Comparison. Figure 5 compares eight editing tasks. In subject removal, several baselines leave artifacts or alter the surrounding box, whereas SDESTEER removes the cake while preserving the box. In subject replacement, Best-of-N produces an ambiguous umbrella–lantern hybrid, while SDESTEER produces the requested paper lantern. The count-change and text-edit examples likewise show more precise edits with less disruption to the background. These cases illustrate the instruction-following and preservation gains reported in Table 1.

Ablation Study. Table 3 isolates the contributions of steering, spatial awareness, and native self-verification. Removing each component reduces the success rate from 82.4% to 77.6%, 74.4%, and 61.1%, respectively. Figure 6 displays the complete method alongside variants without active steering, spatial awareness, or native self-verification. The variants isolate each component of the same editing pipeline for qualitative inspection.

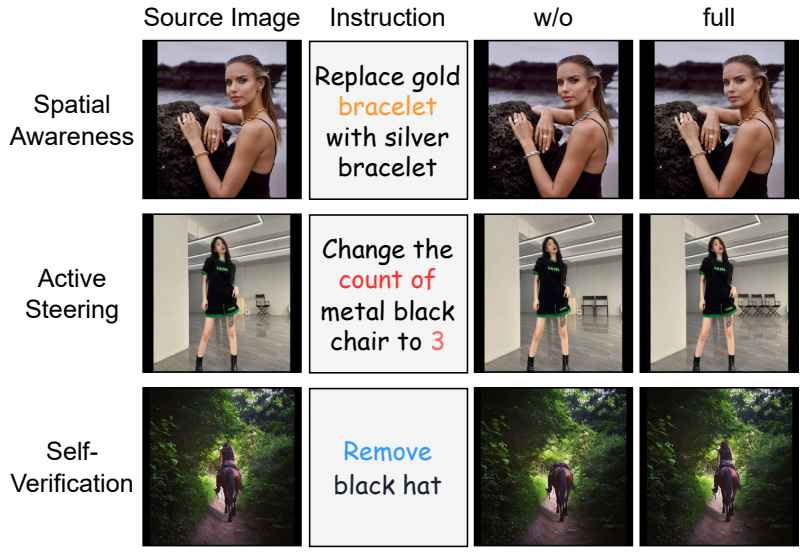

Figure 6: Qualitative comparison of the complete method and three single-component ablations.

5 CONCLUSION

We present SDESTEER, a training-free test-time approach to image editing that combines shared-prior SDE branching, spatially-aware VLM self-verification, and adaptive CFG steering. Its prune-and-steer mechanism uses verification scores to guide reverse SDE trajectories toward instruction adherence while preserving the background, without an external reward model. Performance remains limited by initial branch diversity and VLM perception; extending the method to other diffusion architectures and video editing is future work.

AI USE STATEMENT

In this work, we used generative AI tools (LLM-based assistants) for code development support and manuscript drafting. All AI-assisted content—including text, code, and experimental analysis—was reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Tianyu Chen, Yasi Zhang, Zhi Zhang, Peiyu Yu, Shu Wang, Zhendong Wang, Kevin Lin, Xiaofei Wang, Zhengyuan Yang, Linjie Li, Chung-Ching Lin, Jianwen Xie, Oscar Leong, Lijuan Wang, Ying Nian Wu, and Mingyuan Zhou. Edival-agent: An object-centric framework for automated, fine-grained evaluation of multi-turn editing. *arXiv preprint arXiv:2509.13399*, 2025.

Yingqing Guo, Yukang Yang, Hui Yuan, and Mengdi Wang. Training-free guidance beyond differentiability: Scalable path steering with tree search in diffusion and flow models. *arXiv preprint arXiv:2502.11420*, 2025.

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. *arXiv preprint arXiv:2505.17618*, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *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. In *Advances in Neural Information Processing Systems*, 2025.

Jaihoon Kim, Taehoon Yoon, Jisung Hwang, and Minhyuk Sung. Inference-time scaling for flow models via stochastic generation and rollover budget forcing. In *Advances in Neural Information Processing Systems*, 2025a.

Joowon Kim, Ziseok Lee, Donghyeon Cho, Sanghyun Jo, Yeonsung Jung, Kyungsu Kim, and Eunho Yang. Early timestep zero-shot candidate selection for instruction-guided image editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025b.

Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward over-optimization. In *International Conference on Learning Representations*, 2025c.

Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. Viescore: Towards explainable metrics for conditional image synthesis evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Zheqi Lv, Junhao Chen, Qi Tian, Keting Yin, Shengyu Zhang, and Fei Wu. Multimodal LLM-guided semantic correction in text-to-image diffusion. *arXiv preprint arXiv:2505.20053*, 2025.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Scaling inference time compute for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2523–2534, 2025.

OpenAI. Learning to reason with LLMs. `https://openai.com/index/learning-to-reason-with-llms/`, September 2024. Published September 12, 2024.

Yuta Oshima, Masahiro Suzuki, Yutaka Matsuo, and Hiroki Furuta. Inference-time text-to-video alignment with diffusion latent beam search. In *Advances in Neural Information Processing Systems*, 2025.

Fei Shen, Yutong Gao, Jian Yu, Xiaoyu Du, and Jinhui Tang. IMAGHarmony: Controllable image editing with consistent object quantity and layout. *arXiv preprint arXiv:2506.01949*, 2025.

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. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55810–55827. PMLR, 2025.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-Ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025a.

Jie Wu, Yu Gao, Zilyu Ye, Ming Li, Liang Li, Hanzhong Guo, Jie Liu, Zeyue Xue, Xiaoxia Hou, Wei Liu, Yan Zeng, and Weilin Huang. RewardDance: Reward scaling in visual generation. *arXiv preprint arXiv:2509.08826*, 2025b.

Enze Xie, Junsong Chen, Yuyang Zhao, Jincheng Yu, Ligeng Zhu, Chengyue Wu, 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. *arXiv preprint arXiv:2501.18427*, 2025.

Yi Xin, Siqi Luo, Tianxiang Xu, Qi Qin, Haoxing Chen, Kaiwen Zhu, Zhiwei Zhang, Yangfan He, Rongchao Zhang, Jinbin Bai, Shuo Cao, Bin Fu, Junjun He, Yihao Liu, Yuewen Cao, and Xiaohong Liu. dMLLM-TTS: Self-verified and efficient test-time scaling for diffusion multi-modal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35726–35735, 2026.

Po-Hung Yeh, Kuang-Huei Lee, and Jun-Cheng Chen. Training-free diffusion model alignment with sampling demons. In *International Conference on Learning Representations*, 2025.

Xiangcheng Zhang, Haowei Lin, Haotian Ye, James Zou, Jianzhu Ma, Yitao Liang, and Yilun Du. Inference-time scaling of diffusion models through classical search. *arXiv preprint arXiv:2505.23614*, 2025a.

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 transformers. In *Advances in Neural Information Processing Systems*, 2025b.