REWARDSLIDER: CONTINUOUS IMAGE EDITING VIA REWARD-GUIDED TRAJECTORY CONTROL

Anonymous authors
Paper under double-blind review

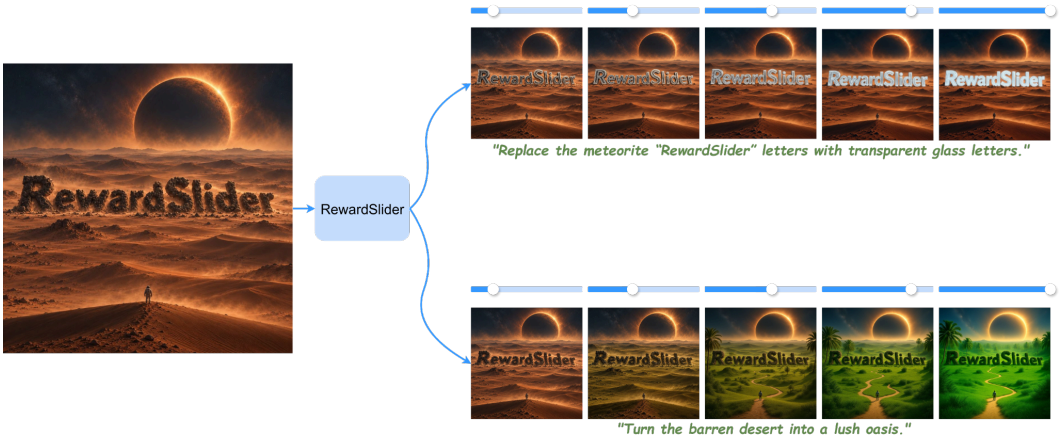

Figure 1: **RewardSlider** provides smooth and continuous strength control for diverse instruction-based edits. A single slider progressively adjusts local appearance or global scene changes while preserving unrelated content, and supports semantic edits ranging from subtle modifications to fully realized transformations.

ABSTRACT

Recent continuous image editing methods provide fine-grained control over instruction strength, but uniform changes to a control signal need not produce uniform changes in the edited image. We present RewardSlider, a reward-guided trajectory calibration framework for continuous image editing. It uses localized velocity-field interpolation to construct a candidate trajectory, then applies output-level feedback to calibrate its effective range and adjust control points for each image–instruction pair. A multi-reward correction targets unintended changes at stronger edit levels without training a separate strength controller. On the 540-example PIE-Bench++ split, RewardSlider obtains the lowest second-order DreamSim score (0.173) and maximum adjacent LPIPS distance (0.210) among the compared methods. SliderEdit records the highest CLIP-Dir (0.447) and final CLIP-T (0.286), while Kontinuous Kontext records the highest monotonicity (0.700).

1 INTRODUCTION

Instruction-based image editors can perform localized and global edits from natural-language prompts Brooks et al. (2023); Sheynin et al. (2024); Black Forest Labs et al. (2025). Interactive use also requires control over edit magnitude: a slider should produce ordered intermediate results whose visual changes track its positions.

Existing continuous editors provide strength controls through learned adapters or explicit conditioning Zarei et al. (2026); Parihar et al. (2026). Training-free approaches instead modify generation dynamics Endo et al. (2026); Li et al. (2026). Across these designs, uniform changes to an internal control value need not yield uniform output changes. The effective response depends on the

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.

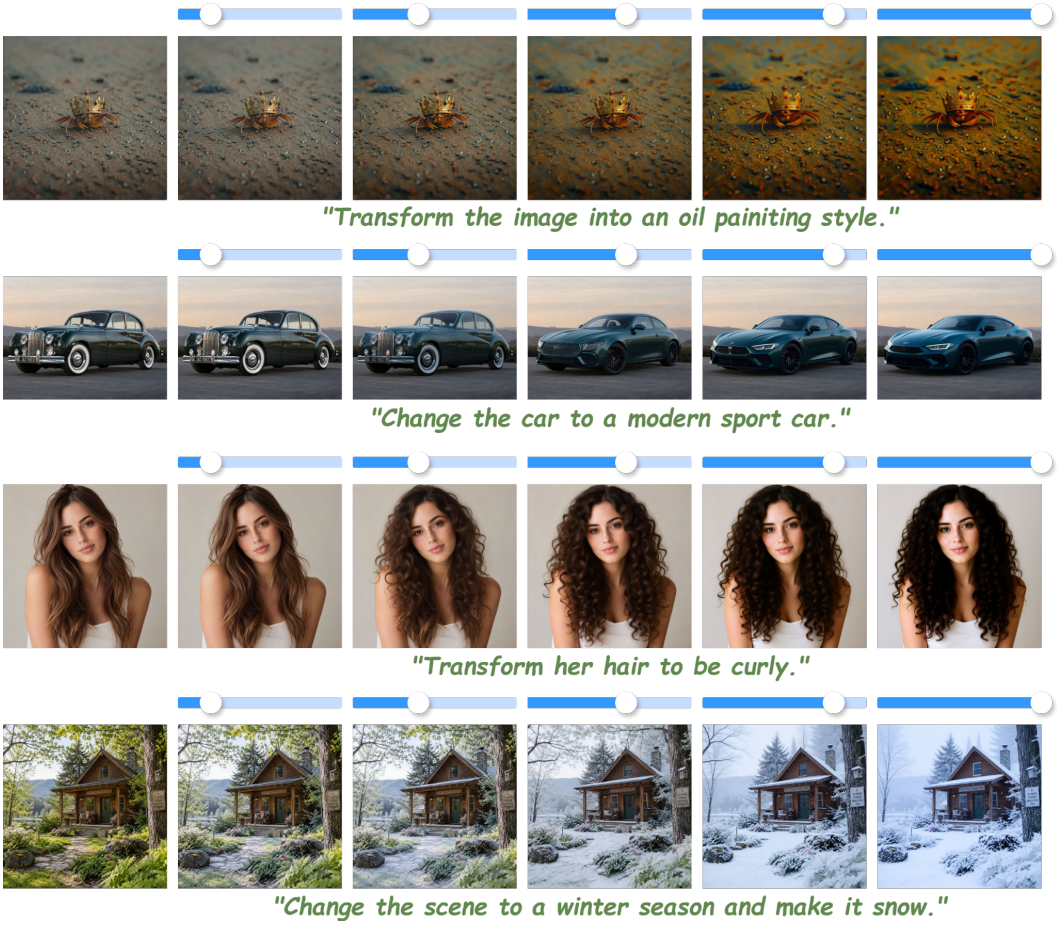

Figure 2: **Qualitative results of continuous edit-strength control with RewardSlider.** Given a source image and an editing instruction, RewardSlider produces smooth and progressively strengthened edits across a wide range of transformations, including style transfer, object transformation, attribute modification, and global scene changes. As the target strength increases from the source state toward the full edit, the intended semantic change is gradually amplified while the overall image structure and instruction-irrelevant content remain consistent.

image, instruction, and sampling trajectory, as also observed in continuous image steering Ekin & Gandelsman (2026).

We present **RewardSlider**, which calibrates a continuous editing trajectory against its generated outputs. It first builds an ordered candidate trajectory with localized velocity interpolation. Output-level perceptual feedback identifies the active strength range and redistributes slider control points. A multi-reward correction then adjusts early trajectory velocities at strengths where the candidate output departs from the intended semantic and preservation targets. The pretrained editor remains frozen.

Our contributions are: (1) formulating continuous editing as output-aware trajectory calibration; (2) using the early, edit-relevant response of the generation dynamics to construct and calibrate instance-specific trajectories; and (3) correcting trajectory states with complementary semantic and preservation rewards. Section 4 evaluates output progression, preservation, runtime, and component effects.

2 Related Work

Continuous image editing. InstructPix2Pix, Emu Edit, and FLUX.1 Kontext provide instruction-conditioned image editing Brooks et al. (2023); Sheynin et al. (2024); Black Forest Labs et al. (2025). Continuous controls have been introduced through semantic adapters and strength conditioning Gandikota et al. (2024); Zarei et al. (2026); Parihar et al. (2026), as well as through inference-time velocity control Endo et al. (2026); Li et al. (2026). RewardSlider uses output-level responses to calibrate a trajectory for each image–instruction pair.

Reward-guided generation. RewardFlow combines differentiable objectives to guide image generation Susladkar et al. (2026); trajectory optimal control optimizes a generation path using terminal reward information Chang et al. (2026). RewardSlider applies this feedback to several strengths along one continuous editing trajectory, calibrating its range and local progression before correcting selected states.

3 Methods

3.1 Flow-Matching Editing Dynamics

RewardSlider uses the frozen FLUX.1 Kontext editor Lipman et al. (2023); Black Forest Labs et al. (2025). Given source image I_{src}, instruction c, and latent state $z(\sigma)$, the model predicts a conditional velocity field. The continuous generation trajectory follows

$$\frac{dz(\sigma)}{d\sigma} = v_\theta(z(\sigma), \sigma; I_{\mathrm{src}}, c), \tag{1}$$

which is numerically integrated over a finite sequence of sampling steps. At step k, let $v_k = v_\theta(z_k, \sigma_k; I_{\mathrm{src}}, c)$ and $\Delta\sigma_k = \sigma_{k+1} - \sigma_k$. A first-order update is

$$z_{k+1} = z_k + \Delta\sigma_k v_k. \tag{2}$$

Since each velocity prediction depends on the current latent, an intervention at step k changes all downstream states. We represent a controlled intervention by

$$\tilde{v}_k = v_k + \Delta v_k, \qquad \tilde{z}_{k+1} = z_k + \Delta\sigma_k \tilde{v}_k. \tag{3}$$

This formulation provides the basis for constructing continuous trajectories and applying localized corrections. The full discretization and intervention analysis are included in Supplementary Sec. 1.

3.2 Spatiotemporal Plasticity of Editing Dynamics

We analyze temporal response by linearizing the discrete dynamics around a native trajectory z_k^*. A perturbation δu_k to the velocity at step k propagates as

$$\delta z_{k+1} = A_k \delta z_k + \Delta\sigma_k \delta u_k, \qquad A_k = I + \Delta\sigma_k J_k, \qquad J_k = \left.\frac{\partial v_\theta}{\partial z}\right|_{z_k^*}. \tag{4}$$

If the intervention is applied only at step k, its first-order effect on the terminal state is

$$\delta z_K = \Delta\sigma_k A_{K-1} \cdots A_{k+1} \delta u_k, \qquad \mathcal{S}(k) = \left\| \frac{\partial z_K}{\partial u_k} \right\|_2 = |\Delta\sigma_k| \, \|A_{K-1} \cdots A_{k+1}\|_2 . \tag{5}$$

Thus the terminal effect depends on how a local intervention propagates through the remaining trajectory. We empirically probe temporal non-uniformity by replacing one ODE transition with an SDE transition and then resuming the deterministic trajectory. Following the ODE-to-SDE probing strategy of Flow-GRPO Liu et al. (2025), earlier branches produce larger terminal pixel RMS variations than later branches, as shown in Fig. 3.

Spatially, we compare the source-preserving velocity $V_{\mathrm{keep},k} = (z_k - z_{\mathrm{src}})/(\sigma_k + \epsilon)$ with the instruction-conditioned velocity $V_{\mathrm{edit},k} = f_\theta(z_k, c, t_k)$. Their element-wise normalized similarity is

$$S_k(i, c') = \frac{|V_{\mathrm{keep},k}(i, c')| + \epsilon}{|V_{\mathrm{keep},k}(i, c')| + \epsilon + |V_{\mathrm{edit},k}(i, c') - V_{\mathrm{keep},k}(i, c')|}. \tag{6}$$

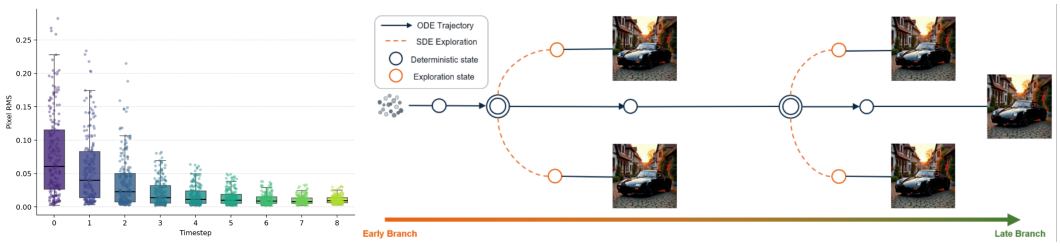

Figure 3: **Timestep-specific plasticity of FLUX-Kontext editing trajectories.** We replace one transition of an otherwise deterministic trajectory with an SDE transition at different timesteps. **Left:** terminal pixel RMS variation decreases markedly from early to late branching points. **Right:** schematic of the probing procedure, illustrating that early stochastic branches induce larger terminal changes than late ones.

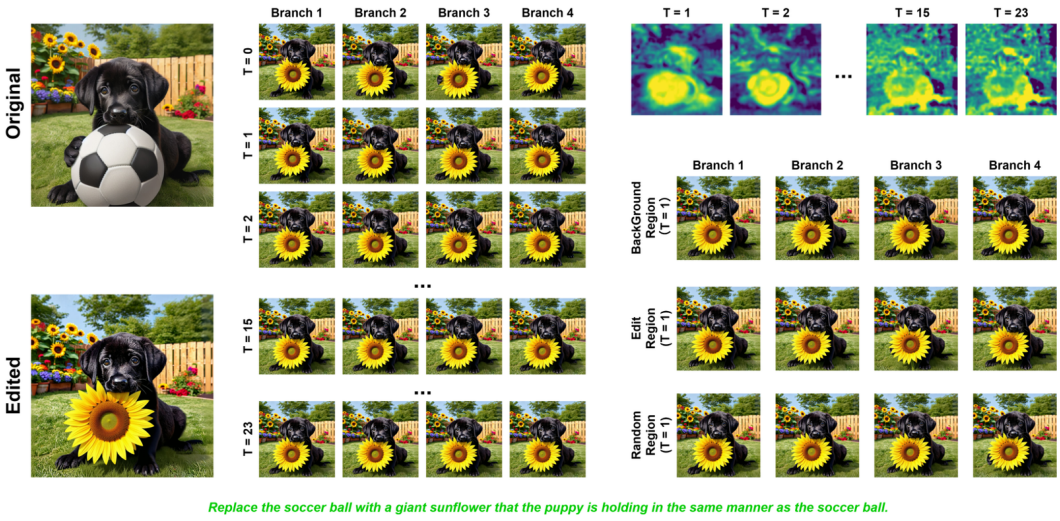

Figure 4: **Spatiotemporal plasticity of FLUX.1 Kontext editing dynamics. Left and center:** stochastic branches introduced at early sampling steps produce substantially larger variations in the terminal edit than those introduced at later steps. **Upper right:** the channel-averaged element-wise velocity discrepancy $m_k(i)$ is concentrated in instruction-relevant regions during the early generation stage. **Lower right:** under matched perturbation magnitudes, interventions within edit-relevant regions induce stronger terminal changes than those applied to background or random regions, revealing spatially localized editing plasticity.

For the first four sampling steps, the preservation and editing masks are

$$M_{\text{keep},k}(i,c') = \mathbb{1}[S_k(i,c') \geq 0.8], \qquad M_{\text{edit},k} = 1 - M_{\text{keep},k}. \tag{7}$$

The controlled velocity interpolates between source-preserving and target-conditioned fields only at editing positions:

$$V_{\alpha,k} = M_{\text{keep},k} \odot V_{\text{keep},k} + M_{\text{edit},k} \odot [(1-\alpha)V_{\text{keep},k} + \alpha V_{\text{edit},k}]. \tag{8}$$

For visualization, we average normalized velocity discrepancy across channels, $m_k(i) = \frac{1}{C}\sum_{c'=1}^{C}(1 - S_k(i,c'))$. At early steps, high-discrepancy positions concentrate in instruction-relevant regions; under matched perturbation magnitudes, interventions in those regions produce larger terminal changes than interventions in background or random regions. Together with the temporal probe, this identifies the early, edit-relevant control window used by RewardSlider. Further probing details are in Supplementary Sec. 1.

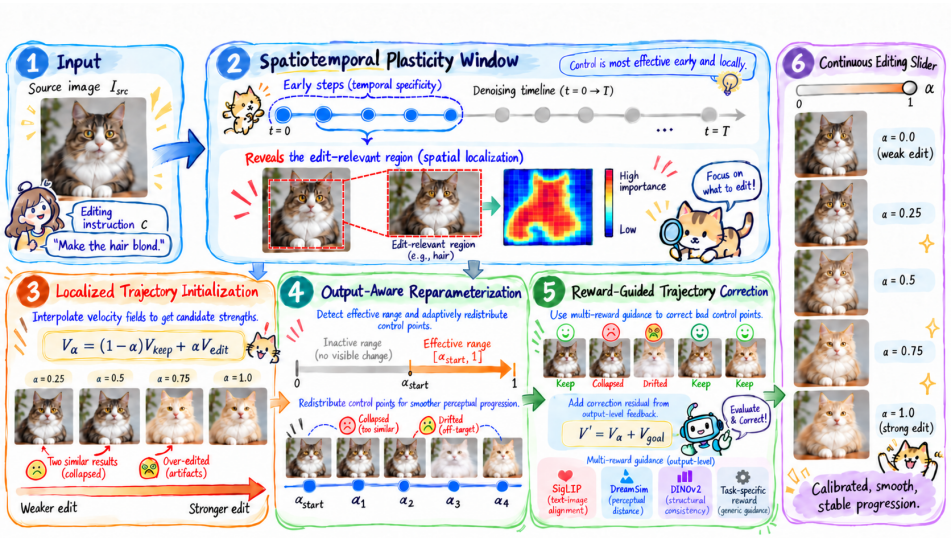

Figure 5: RewardSlider constructs a localized candidate trajectory, calibrates its slider mapping from output-space changes, and applies multi-reward velocity corrections to selected states.

3.3 Reward-Guided Adaptive Trajectory Calibration

We construct a candidate trajectory by interpolating the localized velocity fields in Sec. 3.2. Output-space feedback then calibrates its active range and control-point spacing, followed by reward-guided correction of selected states.

Active range and slider mapping. For an internal strength α, let I_α be its output and $d(\alpha) = d_{\text{DreamSim}}(I_{\text{src}}, I_\alpha)$. We probe $\alpha \in \{0.25, 0.5, 0.75, 1\}$, find the first value with $d(\alpha) > 0.001$, and bisect the inactive–active interval twice. The active endpoint defines α_{start} and the effective range $[\alpha_{\text{start}}, 1]$. Five initial controls use $\beta_i \in \{0, 0.25, 0.5, 0.75, 1\}$ and $\alpha_i = \alpha_{\text{start}} + \beta_i(1 - \alpha_{\text{start}})$.

For neighboring outputs, $g_i = d_{\text{DreamSim}}(I_{\alpha_i}, I_{\alpha_{i+1}})$. We split intervals whose gap exceeds 0.05, reposition controls when adjacent gaps are imbalanced, and remove redundant controls. Refinement uses at most 10 controls and 25 iterations. The calibrated coordinates are $\beta_i = (\alpha_i - \alpha_{\text{start}})/(1 - \alpha_{\text{start}})$; an arbitrary slider value s maps to $\alpha(s)$ by piecewise-linear interpolation through (β_i, α_i). The range search sets the first perceptually active endpoint, while control-point refinement changes how the remaining interval is sampled. Thus, reparameterization adjusts the mapping from slider positions to candidate states; the subsequent residual optimization changes the early velocity field at selected states.

Multi-reward trajectory correction. For calibrated state i, we optimize zero-initialized velocity residuals $V^{(i)}_{\text{goal},k}$ at the first $T_{\text{goal}} = 4$ sampling steps, with the base model frozen:

$$\widetilde{V}^{(i)}_k = V_{\alpha_i,k} + V^{(i)}_{\text{goal},k}, \qquad k < T_{\text{goal}}. \tag{9}$$

The semantic loss penalizes a drop in target-text SigLIP similarity beyond δ from the uncorrected proxy, and preservation combines DreamSim continuity with DINOv2 patch-feature similarity:

$$\mathcal{L}^{(i)}_{\text{sem}} = \left[\max(0, q^{(i)}_{\text{base}} - \delta - q_{\text{SigLIP}}(\hat{I}^{(i)}_0, t))\right]^2, \tag{10}$$

$$\mathcal{L}^{(i)}_{\text{pres}} = d_{\text{DreamSim}}(\hat{I}^{(i)}_0, I^{(i)}_{\text{cont}}) + \lambda_{\text{DINO}} \frac{1}{N} \sum_{j=1}^{N} [1 - \cos(F_j(\hat{I}^{(i)}_0), F_j(I_{\text{safe}}))], \tag{11}$$

$$\mathcal{L}^{(i)}_{\text{goal}} = \lambda_{\text{sem}} \mathcal{L}^{(i)}_{\text{sem}} + \lambda_{\text{pres}} \mathcal{L}^{(i)}_{\text{pres}} + \lambda_{\text{reg}} \,\text{mean}[(V^{(i)}_{\text{goal}})^2]. \tag{12}$$

We use $\lambda_{\text{sem}} = \lambda_{\text{pres}} = \lambda_{\text{DINO}} = 1$, $\lambda_{\text{reg}} = 0.01$, where δ is the allowed decrease from the uncorrected proxy score.



Figure 6: **Qualitative comparison of continuous edit control.** We compare different continuous editing methods across increasing edit strengths for diverse instructions, including scene-level, material, and object appearance edits. Our method produces smooth and progressive edit trajectories while preserving instruction-irrelevant content, whereas competing approaches often exhibit abrupt transitions, insufficient editing, or unintended semantic changes.

To reduce backpropagation through the full trajectory, after the fourth corrected step we estimate the clean latent and decode it as the proxy used for reward evaluation:

$$\hat{z}_0 = z_t - \sigma_t \widetilde{V}_t, \qquad V_{\text{goal}}^{(i)} \to \hat{z}_0^{(i)} \to \hat{I}_0^{(i)} \to \mathcal{L}_{\text{goal}}^{(i)}. \tag{13}$$

The temporal-plasticity experiment in Sec. 3.2 motivates optimizing the early steps. We use Adam with learning rate 10^{-3} for four updates per strength, then generate each final output with the complete 15-step trajectory.

4 Experiments

We evaluate RewardSlider across slider strengths using target-edit alignment, local trajectory consistency, monotonic semantic progression, preservation of unedited content, and computation time. We also isolate activation-range calibration, adaptive control-point placement, and localized reward guidance.

4.1 Evaluation Protocol

Benchmark and instructions. We use the locally hosted PIE-Bench++ Huang et al. (2024) release, which contains 700 source images across nine edit categories. Object addition and deletion are discrete operations without a natural sequence of intermediate strengths; excluding these two 80-image categories leaves 540 examples across seven categories. The annotations provide source and target captions, edit-action metadata, and edit masks. For every method, we form the instruction from the target caption using `Edit the image to match this description:` {`target_caption`} and remove bracketed annotation markers without changing their contents. Because this caption-based instruction differs from the original PIE-Bench protocol, all baseline scores are recomputed on the same local data.

For each image, we evaluate $\mathcal{S} = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$. The source image is used at $s = 0$; methods generate five positive-strength outputs, with $s = 1$ denoting full strength. RewardSlider may make additional internal queries for range calibration and control-point refinement, which are included in runtime. All images are evaluated at 512×512 resolution, with a shared per-example seed when the implementation exposes a seed. We record checkpoints, inference steps, guidance settings, strength mappings, and failed examples for each method.

Table 1: Continuous-editing comparison on the 540-example PIE-Bench++ split. Values are means across examples. All methods are evaluated at six common displayed strengths, including the source at $s = 0$.

Method	CLIP-Dir↑	Final CLIP-T↑	δ_2^{DS} ↓	Mono.↑	Max. LPIPS step↓
VeloEdit	0.220	0.255	0.189	0.660	0.355
FreeMorph	0.184	0.241	0.217	0.658	0.324
Kontinuous Kontext	0.212	0.283	0.361	**0.700**	0.484
SliderEdit	**0.447**	**0.286**	0.225	0.660	0.636
Ours	0.258	0.254	**0.173**	0.664	**0.210**

Compared methods. We compare with VeloEdit Li et al. (2026), which controls editing through velocity interpolation; Kontinuous Kontext Parihar et al. (2026), which conditions on an explicit strength; SliderEdit Zarei et al. (2026), which scales an instruction-aware adapter; and FreeMorph Cao et al. (2025), which interpolates between the source and a full edit. For FreeMorph, its endpoint is generated once per example using the shared FLUX.1-Kontext-dev backbone, and this cost is included in runtime. All methods receive the same source image and caption-based instruction and use their respective inference configurations. A uniform velocity-control variant of RewardSlider is used in the ablations.

Editing and trajectory metrics. CLIP directional similarity (CLIP-Dir) measures alignment between the image direction from source to edit and the text direction from source caption to target caption. We divide the directional similarity by the requested positive strength and average over the five edited positions. Final CLIP-T is the CLIP similarity between the $s = 1$ output and target caption. For trajectory consistency, let d be DreamSim distance and I_0 the source image, followed by edited outputs $I_1, \ldots, I_5$ in increasing strength order. We report the maximum normalized second-order triangle deficit

$$\delta_2^{\mathrm{DS}} = \max_{i=0,\ldots,3} \frac{d(I_i, I_{i+1}) + d(I_{i+1}, I_{i+2}) - d(I_i, I_{i+2})}{\max\{d(I_i, I_{i+2}), \epsilon\}}, \tag{14}$$

where lower values indicate more consistent local transitions. We also report the largest adjacent LPIPS distance, $\max_{i=0,\ldots,4} \mathrm{LPIPS}(I_i, I_{i+1})$, and monotonicity, the fraction of adjacent intervals over which neither target-caption CLIP similarity nor DreamSim distance from the source decreases.

Source preservation. Edit masks identify pixels allowed to change. We evaluate the complement of each mask on the source and intermediate outputs and average LPIPS over $s \in \{0.2, 0.4, 0.6, 0.8\}$; masked L1 is reported as a complementary measure. Full-image masks have no unedited region, leaving 127 examples with a nonempty region outside the edit mask. Preservation scores are aggregated on this mask-valid subset. We report the mean of each metric across examples.

Efficiency protocol. We measure inference on one H20 GPU at batch size one with the same image resolution and five positive-strength outputs used for quality evaluation. GPU operations are synchronized at timing boundaries, and the loaded model is warmed before measurement. Checkpoint loading, disk writes, and offline metric evaluation are excluded. Setup time covers calibration, probing, endpoint generation when required, and optimization before the displayed outputs. Total trajectory time includes setup and all five outputs, including internal probes and reward evaluations. Peak device memory is sampled through NVML every 100 ms during the complete workflow and includes resident reward models. Offline training costs for learned baselines are excluded.

4.2 Continuous-Editing Comparison

On the 540-example split, RewardSlider records the lowest second-order DreamSim (0.173) and maximum adjacent LPIPS (0.210). SliderEdit records the highest CLIP-Dir (0.447) and final CLIP-T (0.286); Kontinuous Kontext records the highest monotonicity (0.700). RewardSlider's corresponding CLIP-Dir, CLIP-T, and monotonicity are 0.258, 0.254, and 0.664.

On the 127 examples with a nonempty unedited region, RewardSlider has the lowest mask-out LPIPS (0.0218) and L1 (0.0120). The second-order DreamSim score summarizes deviations across

Table 2: Source preservation outside the PIE-Bench++ edit mask. The mask-valid pool contains 127 examples from the continuous-editing split. Scores average the four intermediate strengths.

Method	Mask-out LPIPS↓	Mask-out L1↓
VeloEdit	0.0221	0.0124
FreeMorph	0.482	0.0137
Kontinuous Kontext	0.0527	0.0159
SliderEdit	0.0261	0.0153
Ours	**0.0218**	**0.0120**

Table 3: Inference cost on one H20 GPU at 512×512 resolution and batch size one. Times are in seconds. Total time includes all internal method calls needed for five positive-strength outputs. Peak VRAM is measured during that complete workflow.

Method	Setup time↓	Total time↓	Peak VRAM (GB)↓
VeloEdit	12.66	32.13	47.37
FreeMorph	10.25	39.32	40.45
Kontinuous Kontext	24.31	91.82	70.15
SliderEdit	12.64	100.16	36.05
Ours	27.41	337.43	46.79

consecutive triples, whereas maximum adjacent LPIPS isolates the largest single transition. RewardSlider leads on both measures, capturing both trajectory-level regularity and the size of its largest perceptual step. The alignment and monotonicity leaders differ, so these metrics describe complementary aspects of the displayed trajectories.

4.3 Qualitative Results

Figures 2 and 6 show trajectories for style, object, attribute, and scene-level edits. Across these examples, increasing slider strength progressively amplifies the requested change. The comparisons also show differences among methods in transition smoothness, edit magnitude, and preservation of instruction-irrelevant content, complementing the aggregate trajectory and mask-preservation metrics.

4.4 Computational Cost

RewardSlider takes 27.41 seconds for setup and 337.43 seconds for five outputs, with 46.79 GB peak memory. Baseline total times range from 32.13 to 100.16 seconds, and peak memory ranges from 36.05 to 70.15 GB. Its total time is higher than every baseline in this comparison. This measurement includes instance-specific range probes, control-point refinement, reward-guided updates, and generation of all five outputs; setup accounts for 27.41 seconds of the total.

4.5 Ablations and Perceptual Analysis

Table 4: Mechanism ablations on the fixed 90-example stratified subset. The table reports CLIP-Dir, second-order DreamSim, and mask-out LPIPS at six displayed strengths.

Variant	CLIP-Dir↑	δ_2^{DS} ↓	Mask-out LPIPS↓
Uniform velocity control	0.221	0.714	0.0228
- Activation range	0.238	0.586	0.0312
- Adaptive control points	0.249	0.472	0.0261
RewardSlider	0.269	0.286	0.0204
RewardSlider without spatial mask	0.261	0.391	0.0376

A fixed, category-stratified 90-example subset is used for mechanism ablations. RewardSlider improves over uniform velocity control in CLIP-Dir (0.269 vs. 0.221), second-order DreamSim (0.286 vs. 0.714), and mask-out LPIPS (0.0204 vs. 0.0228). The variant without activation-range calibration records 0.238 CLIP-Dir, 0.586 DreamSim, and 0.0312 mask-out LPIPS; without adaptive control points, the corresponding values are 0.249, 0.472, and 0.0261. Removing the spatial mask yields 0.391 DreamSim and 0.0376 mask-out LPIPS. Further probing details are in Supplementary Sec. 1.

5 Conclusion

RewardSlider calibrates continuous image-editing trajectories using output-level perceptual feedback and early-step multi-reward velocity correction. On PIE-Bench++, it records the lowest second-order DreamSim and maximum adjacent LPIPS among the compared methods, while SliderEdit and Kontinuous Kontext lead on CLIP-Dir/final CLIP-T and monotonicity, respectively. RewardSlider also records the lowest preservation distances on the mask-valid subset. Its measured total time for five outputs is 337.43 seconds.

AI Use Statement

Generative AI tools assisted with manuscript organization, translation and editing of prose (including method descriptions), consistency checks between reported results and their textual summaries, literature and reference checking, and preparation of a draft of the architecture figure. The authors reviewed the AI-assisted material and take responsibility for the final text, claims, references, and figures.

References

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. *arXiv preprint arXiv:2506.15742*, 2025. URL https://arxiv.org/abs/2506.15742.

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 (CVPR)*, pp. 18392–18402, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Brooks_InstructPix2Pix_Learning_To_Follow_Image_Editing_Instructions_CVPR_2023_paper.html.

Yukang Cao, Chenyang Si, Jinghao Wang, and Ziwei Liu. FreeMorph: Tuning-Free Generalized Image Morphing with Diffusion Model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18111–18120, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Cao_FreeMorph_Tuning-Free_Generalized_Image_Morphing_with_Diffusion_Model_ICCV_2025_paper.html.

Jinho Chang, Jaemin Kim, and Jong Chul Ye. Training-Free Reward-Guided Image Editing via Trajectory Optimal Control. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/2c3b636b64ca1dfdae3e096e4deeaa42-Abstract-Conference.html.

Yigit Ekin and Yossi Gandelsman. The Unreasonable Effectiveness of Text Embedding Interpolation for Continuous Image Steering. *arXiv preprint arXiv:2603.17998*, 2026. URL https://arxiv.org/abs/2603.17998.

Taichi Endo, Guoqing Hao, and Kazuhiko Sumi. FlowSlider: Training-Free Continuous Image Editing via Fidelity-Steering Decomposition. *arXiv preprint arXiv:2604.02088*, 2026. URL https://arxiv.org/abs/2604.02088.

Rohit Gandikota, Joanna Materzynska, Tingrui Zhou, Antonio Torralba, and David Bau. Concept Sliders: LoRA Adaptors for Precise Control in Diffusion Models. In *Computer Vision – ECCV 2024*, volume 15098 of *Lecture Notes in Computer Science*, pp. 172–188. Springer, 2024. doi: 10.1007/978-3-031-73661-2_10. URL https://doi.org/10.1007/978-3-031-73661-2_10.

Mingzhen Huang, Jialing Cai, Shan Jia, Vishnu Suresh Lokhande, and Siwei Lyu. ParallelEdits: Efficient Multi-Aspect Text-Driven Image Editing with Attention Grouping. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024. doi: 10.52202/079017-0710. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2847043899e1171183ceadf86bdbb280-Abstract-Conference.html.

Zongqing Li, Zhihui Liu, Yujie Xie, Shansiyuan Wu, Hongshen Lv, and Songzhi Su. VeloEdit: Training-Free Consistent and Continuous Instruction-Based Image Editing via Velocity Field Decomposition. *arXiv preprint arXiv:2603.13388*, 2026. URL https://arxiv.org/abs/2603.13388.

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 (ICLR)*, 2023. URL https://arxiv.org/abs/2210.02747.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training Flow Matching Models via Online RL. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, 2025. doi: 10.52202/085713-1362. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/3a10c46572628d58cb44fb705f25cbbf-Abstract-Conference.html.

Rishubh Parihar, Or Patashnik, Daniil Ostashev, Venkatesh Babu Radhakrishnan, Daniel Cohen-Or, and Kuan-Chieh Jackson Wang. Kontinuous Kontext: Continuous Strength Control for Instruction-Based Image Editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 37929–37939, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Parihar_Kontinuous_Kontext_Continuous_Strength_Control_for_Instruction-based_Image_Editing_CVPR_2026_paper.html.

Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu Edit: Precise Image Editing via Recognition and Generation Tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8871–8879, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Sheynin_Emu_Edit_Precise_Image_Editing_via_Recognition_and_Generation_Tasks_CVPR_2024_paper.html.

Onkar Susladkar, Dong-Hwan Jang, Tushar Prakash, Adheesh Juvekar, Vedant Shah, Ayush Barik, Nabeel Bashir, Muntasir Wahed, Ritish Shrirao, and Ismini Lourentzou. RewardFlow: Generate Images by Optimizing What You Reward. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20066–20076, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Susladkar_RewardFlow_Generate_Images_by_Optimizing_What_You_Reward_CVPR_2026_paper.html.

Arman Zarei, Samyadeep Basu, Mobina Pournemat, Sayan Nag, Ryan A. Rossi, and Soheil Feizi. SliderEdit: Continuous Image Editing with Fine-Grained Instruction Control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14430–14439, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zarei_SliderEdit_Continuous_Image_Editing_with_Fine-Grained_Instruction_Control_CVPR_2026_paper.html.