Constrained Look-ahead Guidance for Interference-Aware Flow Editing

Doudou Zhang
ELLIS Institute Finland
Aalto University
doudou.zhang@aalto.fi

Qi Chen
ELLIS Institute Finland
Aalto University
qi.chen.inf@aalto.fi

Abstract

Diffusion and flow models enable powerful generative modeling, but controlled editing remains challenging due to the need to balance source fidelity and edit compliance along the generative trajectory. We propose a principled, inversion-free framework for flow-based image editing from a trajectory centric perspective. Rather than relying on inversion or heuristic guidance, we introduce a one-step look-ahead mechanism that evaluates the effect of candidate updates on both edit and fidelity rewards estimated via denoising, and adjusts the current update based on this evaluation. At each step, our method applies reward-driven corrections guided by the look-ahead evaluation under an explicit fidelity constraint, resulting in stable and coherent trajectory evolution. This look-ahead design mitigates gradient interference between edit and fidelity objectives by anticipating their interaction in future states, leading to improved trade-offs. Experiments demonstrate that the proposed approach achieves stronger semantic edits while better preserving source content compared to existing strong baselines.

1 Introduction

Diffusion [42, 43, 14] and flow models [25, 27] have achieved remarkable success in generative modeling [5] by transforming noise into structured outputs through iterative denoising, enabling applications such as image synthesis [37, 8] and video generation [36]. With the proliferation of large-scale pretrained text–image models [37, 24], there is increasing interest in leveraging them for test-time, text-guided image editing [22, 12, 16], where semantic changes are specified via natural language.

Despite rapid progress, the core challenge of text-guided image editing remains insufficiently understood from a principled standpoint. At its core, editing requires transforming a given sample along a generative trajectory to satisfy two competing objectives: preserving fidelity to the source image while aligning with the target semantics. This naturally motivates a *multi-objective optimization view*. However, existing methods rarely adopt this perspective explicitly or provide principled mechanisms to enforce such trade-offs, often resulting in heuristic and unstable behavior in practice.

Finetuning-based methods [39, 11, 23, 4] achieve strong fidelity and personalization by adapting model parameters, but incur substantial training overhead and reduced flexibility, as adapting to new concepts or instances generally requires additional optimization. Inversion-based approaches [30, 31, 36], such as DDIM inversion [41], map an input image to a latent noise code and reconstruct it through the reverse process. By injecting editing conditions (e.g., text prompts) into this latent, the sampling trajectory can be guided toward target semantics. However, this inversion-based paradigm often yields limited fidelity to the source image [16], primarily due to inaccurate inversion and error accumulation across timesteps, which produce noise trajectories that are not well-suited for controlled editing. To address these limitations, subsequent methods aim to improve inversion accuracy [12, 30, 36]

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

or modify the sampling process, for example by injecting structural cues such as attention maps extracted during inversion [13, 45, 2] or by incorporating test-time optimization [31, 18] to enforce more faithful reconstruction. More recent inversion-free approaches, including FlowEdit [22], bypass inversion by directly modifying the generative vector field. While more efficient, they typically rely on heuristic updates without explicit objectives, which can result in weak or unstable edits.

In this work, we formulate text-guided image editing as a *constrained multi-objective optimization* problem over the generative trajectory. Typical requirement to preserve source fidelity naturally induces a constrained formulation, where edit alignment is optimized subject to such fidelity constraints. Rather than modifying the model or relying on heuristic vector field differences, we perform a local constrained update at each timestep to guide the trajectory. Our key idea is to estimate edit and fidelity rewards via one-step denoising and introduce a *look-ahead* mechanism that evaluates these rewards at the predicted clean endpoints following simulated objective-specific updates. The resulting update incorporates post-intervention reward information to better navigate the edit–fidelity trade-off, while a bounded fidelity correction is applied when necessary to control constraint violations. By anticipating how updates affect future states, this design implicitly resolves the multi-objective conflict and reduces gradient interference between edit and fidelity objectives.

This leads to a well-controlled, inversion-free editing process that achieves strong semantic edits while preserving source content. Our key insight is that trajectory-level control can be realized through a sequence of principled per-step constrained optimizations, rather than global optimization or heuristic guidance. Specifically, our contributions are as follows:

Constrained trajectory-level editing formulation. We formulate text-guided flow editing as a constrained multi-objective control problem over the generative trajectory, where edit compliance is optimized only within a fidelity-feasible region.

Look-ahead surrogate for local Pareto control. Since directly solving the constrained Pareto update at every denoising step is computationally challenging, we derive a one-step look-ahead surrogate that evaluates edit and fidelity rewards at the predicted clean endpoints following simulated objective-specific updates. This yields a locally non-myopic update direction that better navigates the edit–fidelity trade-off by accounting for post-update reward information.

Interference-aware editing behavior. We empirically show that the proposed update improves edit-fidelity gradient alignment, reduces violations of the fidelity constraint, and yields a more favorable edit-fidelity trade-off than greedy edit-first updates. Experiments further demonstrate that our approach enables stronger semantic edits with better source preservation than strong baselines.

2 Related Work

Finetuned-based Editing. Recent works address text-to-image (T2I) generation by fine-tuning pre-trained diffusion or flow models on a few personalized images. Methods such as DreamBooth [39] perform full-model fine-tuning to bind a subject to a unique token, achieving strong identity preservation but requiring costly optimization. DiffusionCLIP [19] performs instance-specific fine-tuning guided by CLIP objectives. Subsequent works improve scalability via multi-concept customization [23] or disentangled identity modeling [4]. To reduce training cost, parameter-efficient approaches such as LoRA [15] were adopted, where SVDiff [11], and T2I-Adapter [32] restrict updates to low-rank or modular components. Alternatively, textual inversion methods [9, 6, 46] optimize token embeddings without modifying model weights, offering lightweight adaptation but limited expressiveness. Despite strong performance, these approaches typically require non-trivial optimization or careful tuning at test time, limiting their practicality for efficient editing.

Inversion-based Editing. Inversion-based methods first map the source image to a noisy latent representation, from which edited samples are generated to maintain consistency with the original input [41, 44]. However, naive inversion often results in insufficient reconstruction fidelity [16]. While several approaches have been proposed to improve inversion accuracy [12, 30, 31, 47], even near-exact inversion on synthetic data does not fully resolve this issue [16]. To address these limitations, a line of work instead intervenes directly in the sampling process by leveraging structural information extracted during inversion and encoded within the model. Attention-control methods [13, 2, 45, 53, 40, 48] preserve spatial layout by manipulating or reusing cross-attention or self-attention maps during generation. However, these methods are often tightly coupled to specific model architectures or

sampling procedures, which limits their adaptability to new settings. Another line of work aims to conduct *test-time optimization* to improve either source fidelity or edit compliance [51, 35, 33, 3]. Null-text Inversion [31] explicitly optimizes the unconditional text embedding during inversion to improve reconstruction. To improve edit compliance, Imagic [18] optimizes text embeddings and fine-tunes the diffusion model, while Pix2Pix-Zero [34] estimates an editing direction in the text embedding space, which can be viewed as an implicit or amortized form of optimization. These approaches highlight the effectiveness of leveraging differentiable objectives at test time. Other methods such as Negative-prompt Inversion [30], Direct Inversion [17], and [12, 10, 38, 29] modify the inversion or sampling trajectory to obtain source-consistent latents without explicit optimization.

Inversion-free Editing. Trajectory guidance methods modify intermediate sampling states using gradient-based updates derived from differentiable objectives, enabling controllable generation without requiring inversion [51, 1, 35, 50]. However, as they generate samples from noise rather than reconstructing source-specific latent representations, they typically lack explicit mechanisms to enforce source fidelity. In contrast, FlowEdit [22] performs inversion-free editing by constructing an ODE that directly maps between the source and target text conditioned distributions, avoiding explicit latent reconstruction.

Unlike approaches mentioned above that treat editing and fidelity separately, we model edit compliance and source fidelity as *coupled, differentiable objectives* along the denoising trajectory, yielding a multi-objective optimization view. As both objectives share the same trajectory, naive aggregation induces gradient conflicts. Building on classical multi-objective optimization methods that mitigate gradient conflicts through projection or Pareto-based updates [7, 52, 26], recent image editing methods have explored strategies to balance editability and fidelity [49, 28].However, these approaches primarily focus on balancing competing objectives without explicitly accounting for the downstream effects of individual updates. In contrast, we introduce a look-ahead formulation that enables non-myopic, per-step constrained optimization, with explicit fidelity constraints for precise editing control.

3 Problem Formulation

Suppose we are given a source content x^{src} and a condition c^{src} describing it. Our goal is to edit x^{src} according to a desired condition c^{edit}. In image editing, x^{src} denotes the latent of the source image, and conditions are typically text prompts. A common approach first inverts x^{src} to a noisy latent via

$$dx_t^{\mathrm{inv}} = v_\theta(x_t^{\mathrm{inv}}, t, c^{\mathrm{src}})\, dt, \quad t \in [t_0, t_T], \quad x_{t_0}^{\mathrm{inv}} = x^{\mathrm{src}}, \tag{1}$$

yielding $x_{t_T}^{\mathrm{inv}}$, and then generates the edited result via

$$dx_t^{\mathrm{gen}} = v_\theta(x_t^{\mathrm{gen}}, t, c^{\mathrm{edit}})\, dt, \quad t \in [t_T, t_0], \quad x_{t_T}^{\mathrm{gen}} = x_{t_T}^{\mathrm{inv}}, \tag{2}$$

where v_θ is a pretrained conditional flow model, x_t^{inv} and x_t^{gen} are latent trajectories.

Impulsive controlled generation. Since flow models are not trained to jointly condition on source images and edit instructions, the generated outputs can deviate substantially from the source, resulting in poor fidelity. This limitation motivates the need for fine-grained control during generation. Leveraging the inherently discretized nature of flow-based sampling, we introduce step-wise control over the latent trajectory and formulate the editing process as an impulsively controlled dynamics.

$$dx_t^{\mathrm{gen}} = v_\theta(x_t^{\mathrm{gen}}, t, c^{\mathrm{edit}})\, dt + \sum_{k=1}^{T} u_k\, \delta(t - t_k). \tag{3}$$

Between two adjacent timesteps (t_{k-1}, t_k), the trajectory follows the pretrained edit-conditioned dynamics. At discrete timesteps, we apply an impulsive control u_k to the latent state.

Multi-objective control at each impulse. Control updates generally affect both edit compliance and source fidelity. Since realistic image editing requires preserving instance-specific details, excessive edits can make the result indistinguishable from unconditional generation. We therefore formulate editing as a constrained, trajectory-level multi-objective optimization problem, enforcing a minimum fidelity requirement ϵ during generation. Let $r_{\text{edit}}(x, t)$ and $r_{\text{fid}}(x, t)$ be differentiable rewards for edit quality and fidelity. At each timestep t_k, we update the latent via

$$\tilde{x}_k = x_k^{\mathrm{gen}} + u_k, \tag{4}$$

where u_k is obtained by solving the following constrained local multi-objective control problem:

$$u_k^\star \in \mathrm{Pareto}\left\{(r_{\mathrm{edit}}(x_k^{\mathrm{gen}} + u_k, t_k), r_{\mathrm{fid}}(x_k^{\mathrm{gen}} + u_k, t_k)) : r_{\mathrm{fid}}(x_k^{\mathrm{gen}} + u_k, t_k) \geq \epsilon\right\}, \tag{5}$$

where the definition of Pareto set is deferred to Sec. A. Here, the fidelity constraint defines the feasible region, while the Pareto objective characterizes the local edit–fidelity trade-off within that region. Directly solving Eq. (5) at every sampling step is impractical. We therefore approximate this local multi-objective control problem using tractable surrogate update rules, which are introduced in Sec. 4.

4 Methodology

Building on the preceding problem formulation, we detail its practical implementation by defining (a) the inversion, inversion-free initialization, and generation procedures, and (b) the differentiable rewards for editing compliance and source fidelity.

Inversion, inversion-free, and generation trajectories. Let $\{t_k\}_{k=0}^T$ be a discretization with $t_0 < \cdots < t_T$. We denote by x_k^{inv} the inversion state at time t_k. A forward Euler discretization of equation 1 gives

$$x_{k+1}^{\mathrm{inv}} = x_k^{\mathrm{inv}} + (t_{k+1} - t_k)\, v_\theta\big(x_k^{\mathrm{inv}}, t_k, c^{\mathrm{src}}\big), \quad x_0^{\mathrm{inv}} = x^{\mathrm{src}}. \tag{6}$$

Alternatively, an inversion-free initialization constructs the noisy state as

$$x_k^{\mathrm{noisy}} = (1 - t_k)\, x^{\mathrm{src}} + t_k\, N_k, \quad N_k \sim \mathcal{N}(0, I). \tag{7}$$

For editing, we initialize $x_T^{\mathrm{gen}} = x_T^{\mathrm{inv}}$ or $x_T^{\mathrm{gen}} = x_T^{\mathrm{noisy}}$, depending on the initialization strategy. The reverse-time explicit Euler discretization of Eq. (2) is:

$$x_{k-1}^{\mathrm{gen}} = x_k^{\mathrm{gen}} + (t_{k-1} - t_k)\, v_\theta\big(x_k^{\mathrm{gen}}, t_k, c^{\mathrm{edit}}\big), \tag{8}$$

yielding x_0^{gen} as the edited image x^{edit}. At each timestep t_{k-1}, we treat the current latent x_{k-1}^{gen} as the optimization variable and apply differentiable guidance updates for desired control.

Reward definition via clean-state approximation. Fidelity and edit compliance are defined on the underlying clean content, which is not directly accessible during intermediate steps. We therefore approximate the clean latent at each timestep using a single-step denoising estimate:

$$\hat{x}_0(x, t, c) := x - t\, v_\theta(x, t, c). \tag{9}$$

Although approximate, this estimate is sufficient for computing informative gradients, as most reward functions operate on clean images.

We define **edit reward** as a similarity between the predicted clean content and the edit condition:

$$r_{\mathrm{edit}}(x, t) := \mathrm{Sim}\big(\hat{x}_0(x, t, c^{\mathrm{edit}}),\, c^{\mathrm{edit}}\big). \tag{10}$$

In practice, $\mathrm{Sim}(\cdot, \cdot)$ can be instantiated using different CLIP-based objectives, including text-image similarity, directional similarity, or contrastive differences, as summarized in Appendix C.

We measure **fidelity reward** by the reconstruction error between the predicted clean content under the source condition and the original source latent:

$$r_{\mathrm{fid}}(x, t) := -\big\|\hat{x}_0(x, t, c^{\mathrm{src}}) - x^{\mathrm{src}}\big\|_2^2. \tag{11}$$

This encourages the generated trajectory to remain close to the source content while allowing controlled deviations for editing.

Remark 4.1. Fidelity is a proxy, not the target. Importantly, r_{fid} is only an imperfect surrogate for perceptual or semantic fidelity. Using the original image as the reference is not fully justified when editable and preservable content are not explicitly separated: perfect fidelity would imply no edit at all. Thus, treating fidelity as a symmetric objective conflates the desired constraint of preserving what should remain unchanged with an imperfect reward proxy.

Balancing edit alignment and source fidelity. Edit alignment and source fidelity are inherently conflicting objectives: perfect fidelity leads to trivial reconstruction with no semantic modification. We therefore formulate image editing as a constrained multi-objective optimization problem over

the generative trajectory, where edit rewards are maximized while preserving source consistency through explicit fidelity constraints. A direct linear scalarization $\max_x \; r_{\text{edit}}(x,t) + \lambda\, r_{\text{fid}}(x,t)$ is often brittle in practice, since the optimal trade-off parameter λ varies significantly across timesteps due to the changing geometry of the denoising trajectory. Moreover, edit and fidelity gradients may conflict locally, resulting in unstable or myopic updates. A simple constrained alternative first applies an **edit-first** update $x \leftarrow x + \gamma\, \nabla_x r_{\text{edit}}(x,t)$, which is accepted as long as the fidelity constraint $r_{\text{fid}}(x_k^{\text{gen}}, t_k) \geq \varepsilon$ remains satisfied. If the constraint is violated, approximate feasibility is pursued through projected correction steps, $x \leftarrow x + \eta\, \nabla_x r_{\text{fid}}(x,t)$, repeated for at most J iterations. The finite correction budget bounds computation but does not guarantee exact feasibility at every inner iteration. This constrained refinement acts as a local trust-region mechanism, preventing excessive deviation from the source image while still allowing flexible semantic edits.

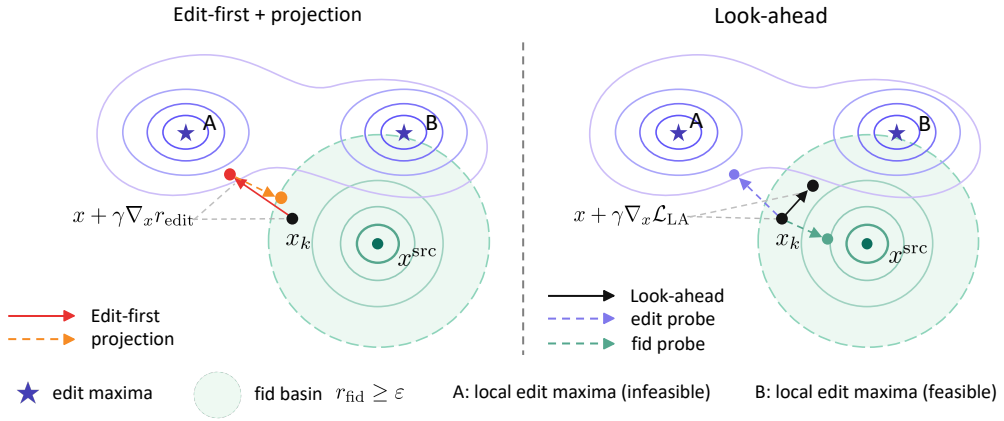

Figure 1: **Mechanism comparison.** *Left*: edit-first update moves along g_{edit} but may violate fidelity constrain and need projects back to the fidelity basin. *Right*: the look-ahead update uses edit and fidelity probes to evaluate $\nabla_x \mathcal{L}_{\text{LA}}$ before committing an update, steering the trajectory toward the feasible edit maximum inside the fidelity basin.

This strategy remains purely local and greedy, as each update is determined only by the current state. In diffusion and flow trajectories, such local updates can accumulate over time, causing semantic drift or source-fidelity degradation in later steps. Analogously, existing multi-objective optimization (MOO) methods, such as MGDA [7] and PCGrad [52], are also designed to resolve gradient conflicts at the current optimization state. However, this local view is insufficient for flow-based image editing, where each update is propagated through the remaining generation trajectory before affecting the final image. As a result, a locally conflict-free direction may still induce an undesirable future trajectory.

Look-ahead guidance. To address this issue, we approximate the constrained MOO update with a look-ahead strategy that anticipates the effects of candidate updates on the predicted clean endpoints, followed by a bounded fidelity correction when necessary. Specifically, we first construct two look-ahead states

$$x_{\text{edit}}(x) = x + \alpha \nabla_x r_{\text{edit}}(x,t), \qquad x_{\text{fid}}(x) = x + \beta \nabla_x r_{\text{fid}}(x,t), \tag{12}$$

which represent edit-oriented and fidelity-oriented updates, respectively. We then define the look-ahead objective

$$\mathcal{L}_{\text{LA}}(x) := r_{\text{edit}}(x_{\text{edit}}(x), t) + \lambda r_{\text{fid}}(x_{\text{fid}}(x), t), \tag{13}$$

and update the latent using

$$x \leftarrow x + \gamma \nabla_x \mathcal{L}_{\text{LA}}(x). \tag{14}$$

Importantly, each reward in Eq. (13) is evaluated on the condition-specific predicted clean endpoint recomputed from the corresponding perturbed latent. Unlike direct scalarization, which combines reward gradients at the current latent, our look-ahead objective accounts for how estimated updates alter the predicted downstream rewards before committing to an actual update. This incorporates local curvature information arising from both the reward functions and the denoising map, yielding a locally non-myopic update without explicitly unrolling additional denoising steps. We provide further analysis in Appendix F.

Full generation-phase algorithm. Starting from the initial latent, each generation timestep first applies an edit-conditioned Euler step and then performs p inner-loop trust-region refinements. At each refinement step, either the edit-first or look-ahead update rule proposes a latent update by evaluating rewards on the predicted clean state $\hat{x}_0$ and its decoded image. The fidelity constraint then corrects updates that induce excessive perceptual drift from the source appearance. The complete procedure is given in Algorithm 1.

5 Experiments

We structure the experiments around four research questions. **RQ1**: How effective is our method for real-image editing versus baselines? **RQ2**: Does gradient conflict between edit and fidelity objectives arise during generation, and can look-ahead mitigate it? **RQ3**: Does our method align with its intended reward-space formulation? **RQ4**: How does the fidelity threshold ε affect the edit–fidelity trade-off? We address them via main comparisons, gradient-conflict analysis, reward-space trajectory analysis, and ε-sensitivity analysis.

5.1 Experimental Setup

We evaluate our algorithm[1] and baselines on PIE-Bench v1 [17], a real-image editing benchmark spanning 10 edit categories, including object replacement, addition, deletion, attribute editing, background modification, and style transfer. For fair comparison, we report results on this public benchmark, where all baselines generate valid outputs under a unified FLUX-based evaluation pipeline. When applicable, all methods use FLUX.1-dev [24] as the backbone.

We compare against eight baselines. DDIM Inversion [41] inverts the source image to a noise latent via ODE and regenerates it with the edit prompt. TABA [44] progressively edits the image by alternating forward noising and edit-prompt denoising. RF-Inversion [38] uses a stochastic ODE for inversion and injects source features during generation to preserve structure. RF-Solver-Edit [48] maintains fidelity by injecting V-values from the source branch into the edit branch. FlowEdit [22] is inversion-free and shifts the trajectory using the velocity difference between source and edit conditions. FlowChef-Edit [35] steers the rectified-flow trajectory with gradient guidance and attention-based region masks. FlowAlign [20] regularizes the editing trajectory toward the source flow. ITOC [3] inverts the source image and optimizes the full trajectory control with a terminal reward via adjoint states. We adapt the official implementations of FlowAlign and ITOC to the same FLUX backbone. DDIM Inversion and TABA are also adapted to flow models in our evaluation. When available, all baselines use the default hyperparameters from their official repositories. For fairness, we report the ConceptAttention (CA) variant of FlowChef-Edit, which uses masks automatically extracted by ConceptAttention rather than human annotations.

For our method, we use inversion-free initialization, constructing the initial noisy state by forward flow without separate inversion. We run FLUX.1-dev for 28 generation steps and apply multi-objective test-time optimization at every denoising step. Each latent is refined with three inner optimization iterations before advancing to the next timestep. The edit objective is a CLIP-contrastive reward. During generation, edit and fidelity gradients are computed on a single-step clean-latent prediction x_0 from the current noisy latent. Source preservation uses a latent-ℓ_2 fidelity measure with threshold $\varepsilon = -0.1$. When the fidelity constraint is violated, we apply correction updates with step size $\eta = 5.0$, for at most three correction steps per denoising step. We use ℓ_2-normalized gradients and set the edit probe, fidelity probe, and committed-update step sizes to $\alpha = \beta = \gamma = 5.0$, with fidelity weight $\lambda = 1$. Full sensitivity analyses are provided in Appendix F.4.

Following prior work [17, 35], we evaluate source preservation on unedited regions using structure distance, PSNR, LPIPS, MSE, and SSIM. These metrics capture complementary fidelity aspects: structure distance measures spatial-layout preservation; PSNR and MSE assess pixel-level reconstruction; LPIPS measures learned perceptual similarity; and SSIM evaluates luminance, contrast, and structural consistency. Better preservation corresponds to higher PSNR and SSIM, and lower structure distance, LPIPS, and MSE. For edit quality, we report CLIP similarity between the edited image and target prompt, computed on both the whole image and edited region. Whole-image CLIP

[1] Code is available at `https://github.com/aalto-icl/ConstrainedLookaheadGuidance`.

Table 1: Quantitative comparison on the PIE-Bench [17]. Our method achieves competitive preservation metrics while also obtaining the best global target-prompt alignment. Best results are in **bold** and second-best results are underlined.

Method	Structure ↓	Background Preservation				CLIP Similarity	
		PSNR ↑	LPIPS ↓	MSE ↓	SSIM ↑	Whole ↑	Edited ↑
DDIM Inversion [41]	0.1229	12.7867	0.3578	0.0763	0.6025	26.7718	23.9214
TABA [44]	0.1233	12.8892	0.3496	0.0744	0.6056	27.4278	**24.2488**
RF-Inversion [38]	0.0561	19.1570	0.2290	0.0171	0.6600	26.1720	23.3210
RF-Solver-Edit [48]	0.0249	24.4093	0.1149	0.0057	0.8446	25.6668	22.7751
FlowChef-Edit [35]	0.0395	24.3100	0.1239	–	0.7960	25.3200	22.3400
FlowEdit [22]	0.0276	22.0007	0.1068	0.0094	0.8426	26.0726	23.1487
FlowAlign [20]	0.0979	14.6995	0.2599	0.0469	0.6977	27.0154	23.2178
ITOC [3]	0.0188	**26.5574**	0.1125	**0.0042**	**0.8513**	27.6348	21.6832
Ours	**0.0176**	25.0050	**0.0957**	0.0045	0.8433	**28.6262**	23.7916

measures global text-image alignment, while edited-region CLIP assesses whether the intended modification is realized in the target area.

5.2 Quantitative Comparison

Table 1 summarizes the quantitative comparison. Our method achieves the best structure distance and LPIPS on unedited regions, as well as the best whole-image target-prompt alignment. ITOC [3] attains the best PSNR, MSE, and SSIM, but at substantially lower edited-region alignment, and our method ranks second on PSNR and MSE. For edited-region CLIP similarity, it ranks third behind TABA [44] and DDIM Inversion [41]. However, these methods lie at the high-edit extreme of the trade-off, improving local alignment at the cost of much worse preservation. In contrast, our method reaches a more favorable frontier point, strongly preserving unedited content while surpassing all baselines in global edit alignment.

5.3 Qualitative Comparison

As shown in Figure 2, our method produces edits that are more faithful to the target instructions while better preserving the source structure and visual realism than competing methods. In the deer-addition example, our method inserts the deer at an appropriate location while leaving the surrounding trees, flowers, and road almost unchanged. Competing methods, by contrast, nearly regenerate a new forest background. For the object-attribute editing task, our method changes the bird's feathers to green precisely. FlowEdit [22] also applies the color change, but the bird's shape is altered substantially. Similarly, in the object-replacement example, baselines such as RF-Solver-Edit [48] and RF-Inversion [38] fail to preserve the background leaves, whereas our method generates a butterfly with the correct color while keeping the leaves unchanged. In the facial-expression editing example, most competing methods almost replace the child entirely. Our method instead achieves the semantic edit through subtle changes around the eyes and face, without the background corruption and identity drift observed in other methods.

5.4 Diagnostic Analysis

5.4.1 Gradient Conflict along the Generation Trajectory

We analyze the gradient conflict between edit and fidelity objectives along the generation trajectory, and show that the proposed look-ahead mechanism improves gradient alignment. At generation step k, we measure gradient alignment by the cosine similarity $c_k = \cos(\nabla_x r_{\text{edit}}(x_k, t_k), \nabla_x r_{\text{fid}}(x_k, t_k))$, where x_k is the latent at timestep t_k. Figure 3(a) reports the mean per-step gradient alignment $\frac{1}{T}\sum_k c_k$ as a function of ε. The look-ahead proposal consistently achieves higher alignment than the edit proposal across all fidelity thresholds.

When the edit gradient and fidelity gradient are better aligned, the proposed update is less likely to cross the fidelity threshold ε. Figure 3(b) confirms this: the look-ahead proposal incurs substantially fewer violations per sample than the edit proposal across all ε values, particularly at loose thresholds.

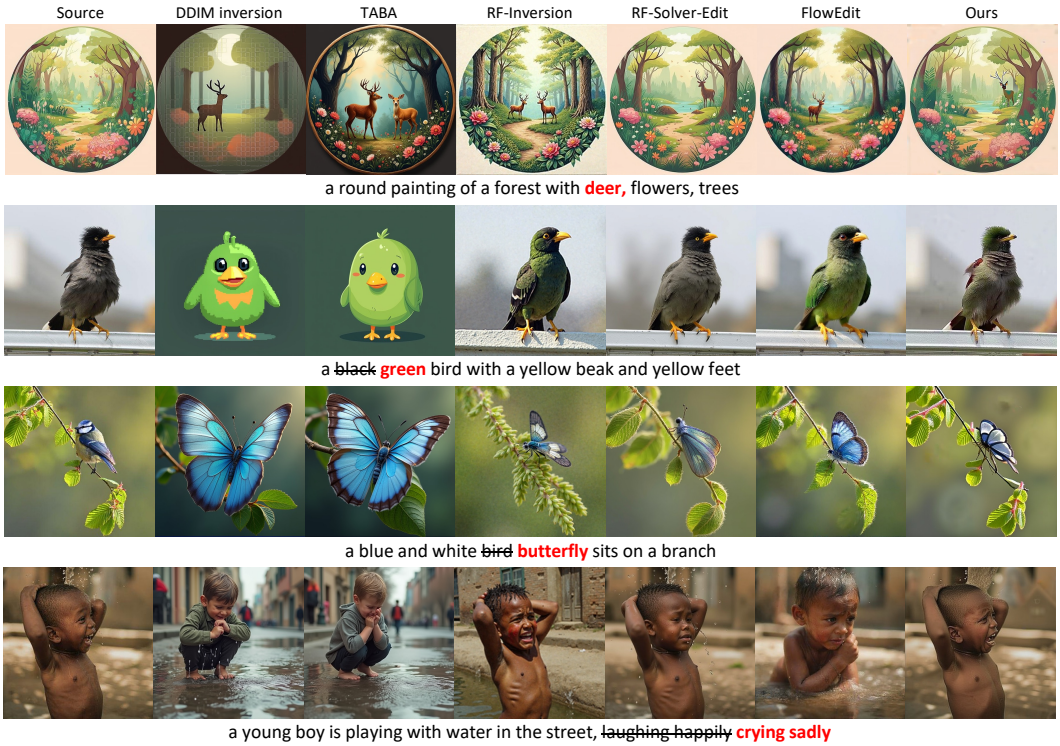

Figure 2: **Qualitative comparison on representative PIE-Bench edit categories.** Our method consistently completes the requested edit while preserving unedited content.

Figure 3(c) shows that the $(r_{\text{fid}}, r_{\text{edit}})$ operating points of the two rules trace nearly the same curve across the ε sweep. This says that the proposed look-ahead mechanism reduces the number of fidelity violation without sacrificing edit quality. We also observe that the look-ahead operating points saturate around $r_{\text{fid}} \approx -0.5$ regardless of how loose ε becomes, unlike edit-first which continues to degrade fidelity. This is because the look-ahead gradient inherently couples both objectives at every step, preventing the update direction from becoming purely edit-driven even under a slack constraint.

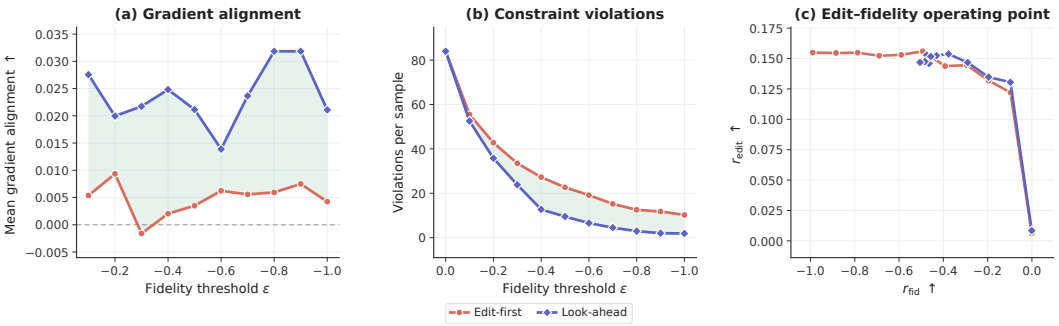

Figure 3: **Gradient conflict and look-ahead analysis** across PIE-Bench samples. *(a)* Mean per-step alignment $\frac{1}{T}\sum_k c_k$ vs. ε: the look-ahead proposal achieves consistently higher alignment than the edit proposal. *(b)* Violations per sample vs. ε: the look-ahead rule triggers substantially fewer fidelity violations, especially at loose thresholds. *(c)* $(r_{\text{fid}}, r_{\text{edit}})$ operating points: both rules trace nearly the same curve, confirming that fewer violations do not degrade edit quality.

5.4.2 Reward-Space Trajectories under Different Update Rules

Figure 4 compares four generation-time update rules in reward space: the base trajectory without optimization, edit-only updates, fidelity-only updates, and our proposed method. The comparison

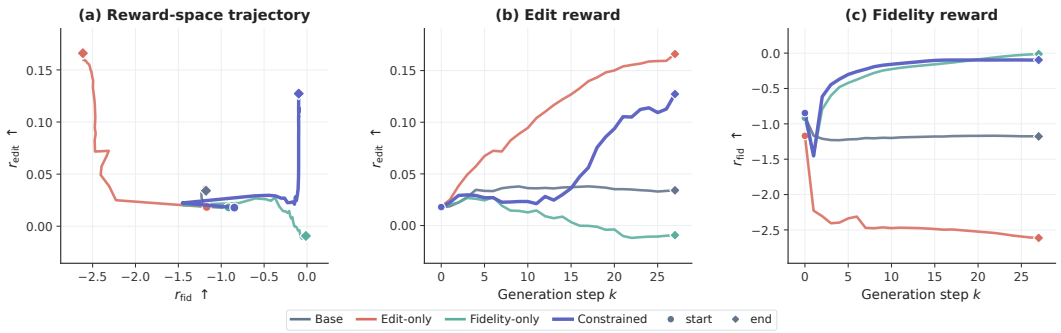

Figure 4: **Reward-space trajectories under different generation-time update rules.** *(a)* Trajectories in the $(r_{\text{fid}}, r_{\text{edit}})$ plane; *(b)* edit reward; *(c)* fidelity reward over generation steps. Edit-only updates improve r_{edit} aggressively but severely degrade fidelity; fidelity-only updates preserve source consistency but suppress editing. Our method reaches a balanced trajectory, improving edit reward while keeping fidelity degradation controlled.

makes the role of the proposed method explicit. Edit-only updates improve r_{edit} most aggressively, but do so by driving the trajectory to a severely degraded fidelity regime. Fidelity-only updates move in the opposite direction: they recover source consistency, but suppress semantic editing and eventually reduce edit reward. The base trajectory rises slightly in the direction of editing reward, mainly because the generated trajectory is controlled by the target condition, but decreases in the dimension of fidelity reward.

Our method follows a more balanced path. As shown in Fig. 4(a), it moves the trajectory toward substantially higher edit reward without collapsing fidelity to the extremes reached by edit-only updates. Panels (b) and (c) make the same behavior visible over time: r_{edit} increases steadily, while r_{fid} initially drops as semantic changes emerge and then stabilizes at a controlled level rather than drifting arbitrarily. This behavior is consistent with our formulation. The method performs edit-first optimization under a fidelity threshold.

5.4.3 The Edit–Fidelity Trade-Off under Different Fidelity Thresholds

Figure 5 illustrates how the fidelity threshold ε shapes the reward-space trajectory across PIE-Bench samples under the edit-first proposal rule used in all runs. The left panel traces the $(r_{\text{fid}}, r_{\text{edit}})$ operating point of each run as ε varies. The curve sweeps continuously from the upper-left (small ε, loose constraint, high edit reward) to the lower-right (large ε, strict constraint, high fidelity reward), forming an empirical curve parameterized directly by ε.

Two qualitative regimes are visible. For $\varepsilon \leq -0.5$, the edit reward saturates: loosening the constraint further provides diminishing returns in edit quality while fidelity continues to degrade. Above $\varepsilon \approx -0.25$, the edit reward decreases gradually as the fidelity projection increasingly overrides the edit gradient in the low-noise stage of generation. At $\varepsilon = 0$ the constraint is so strict that the method reduces to approximate reconstruction, and edit reward collapses to near zero.

These observations do not prescribe a single optimal ε; the appropriate setting depends on the task. For image editing, fidelity and edit compliance are both essential: a value that preserves unedited regions while still completing the requested semantic change. We find that $\varepsilon = -0.1$ on the latent-ℓ_2 fidelity metric achieves this balance in practice. Panels (b) and (c) of Figure 5 show that this setting produces a steady increase in r_{edit} through the generation trajectory while keeping r_{fid} above the constraint boundary, which is consistent with the PIE-Bench preservation results reported in Table 1.

5.4.4 Computational Cost

Table 2 reports per-image runtime and peak GPU memory on a single NVIDIA H200 GPU. Methods that perform reward-gradient optimization, including ITOC [3] and ours, incur higher costs than baselines with fixed inversion and generation procedures. Our additional overhead comes from evaluating both objectives and applying iterative fidelity corrections. As detailed in Appendix F.3, linearized projection reduces runtime to 139.4 s. Improving efficiency remains important future work.

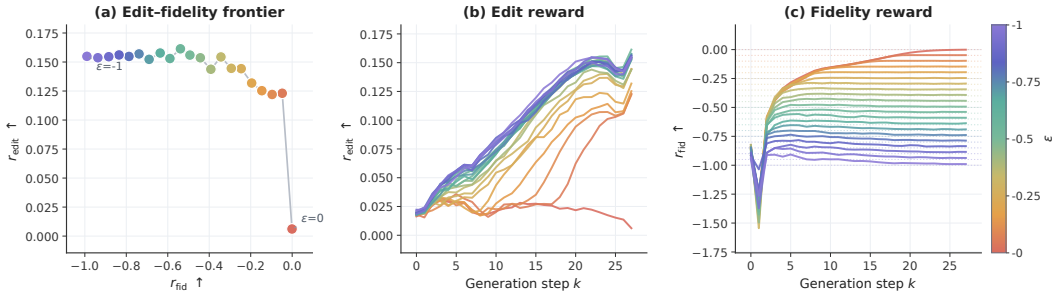

Figure 5: **Effect of the fidelity threshold ε on the edit–fidelity trade-off.** *(a)* $(r_{\text{fid}}, r_{\text{edit}})$ per ε: the connected curve forms an empirical curve parameterized directly by ε. *(b,c)* Edit and fidelity reward over generation steps; dotted lines in *(c)* mark the constraint boundary.

Table 2: Inference runtime and peak GPU memory.

Method	Runtime / image (s) ↓	Peak memory (GiB) ↓
FlowEdit [22]	**7.00**	34.18
TABA [44]	7.80	**32.05**
DDIM Inversion [41]	8.08	**32.05**
RF-Solver-Edit [48]	16.45	32.59
RF-Inversion [38]	25.90	35.72
ITOC [3]	139.25	40.93
Ours	185.88	55.82

Table 3: Edit-first versus look-ahead under the same per-image projection budget.

Update rule	Structure ↓	PSNR ↑	LPIPS ↓	MSE ↓	SSIM ↑	CLIP Similarity (Whole) ↑	CLIP Similarity (Edited) ↑
Edit-first	0.0138	26.2152	0.0841	0.0033	0.8258	26.9845	**22.0966**
Look-ahead	**0.0091**	**27.2950**	**0.0751**	**0.0026**	**0.8476**	**27.2370**	22.0130

5.4.5 Ablation Studies

We summarize key ablations below; Appendix F reports full results and a user study.

Update rule. Table 6 shows that edit-first and look-ahead reach nearly identical endpoints on the full PIE-Bench under unrestricted correction, suggesting that the constrained framework accounts for most of the overall gain. Under a matched per-image projection budget, Table 3 shows that look-ahead improves every preservation metric, reaching a favorable edit–fidelity operating point with less reliance on corrective projections.

Hyperparameters. Increasing p or J beyond 3 adds cost for marginal gains (Table 8). Performance is stable over a $16\times$ range of λ and probe distances $\alpha, \beta \in [1, 10]$ (Table 9). We use $p = J = 3$ and $\lambda = 1$ as practical defaults.

Rewards and initialization. ImageReward and HPS v2.1 yield a similar edit–fidelity balance to CLIP-contrastive guidance (Table 11). Inversion-free initialization improves every preservation metric over inversion (Table 10).

6 Conclusion

We have presented a principled, inversion-free framework for text-guided image editing that formulates the problem as constrained multi-objective optimization over the generative trajectory. The framework evaluates candidate interventions through their predicted clean endpoints and enforces an explicit source-fidelity constraint. The look-ahead update improves gradient alignment and reduces reliance on corrective projections, while the constraint exposes a controllable edit–fidelity operating point. Experiments demonstrate stronger source preservation and target alignment than existing inversion-based and inversion-free methods, with the trade-off of additional inference-time computation.

Acknowledgments and Disclosure of Funding

Doudou Zhang and Qi Chen are supported by research funding from the ELLIS Institute Finland and the Department of Computer Science at Aalto University. We acknowledge CSC-IT Center for Science, Finland, for providing access to the supercomputers Roihu and LUMI, owned by the European High Performance Computing Joint Undertaking (EuroHPC JU) and hosted by CSC Finland in collaboration with the LUMI consortium. We also acknowledge the computational resources provided by the Aalto Science-IT project through the Triton cluster.

References

[1] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 843–852, 2023.

[2] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. MasaCtrl: Tuning-Free Mutual Self-Attention Control for Consistent Image Synthesis and Editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22560–22570, 2023.

[3] Jinho Chang, Jaemin Kim, and Jong Chul Ye. Training-Free Reward-Guided Image Editing via Trajectory Optimal Control. *arXiv preprint arXiv:2509.25845*, 2026.

[4] Hong Chen, Yipeng Zhang, Simin Wu, Xin Wang, Xuguang Duan, Yuwei Zhou, and Wenwu Zhu. Disenbooth: Identity-preserving disentangled tuning for subject-driven text-to-image generation. *arXiv preprint arXiv:2305.03374*, 2023.

[5] Qi Chen, Jierui Zhu, and Florian Shkurti. Generalization in VAE and Diffusion Models: A Unified Information-Theoretic Analysis. In *The Thirteenth International Conference on Learning Representations*, October 2024.

[6] Giannis Daras and Alexandros G Dimakis. Multiresolution textual inversion. *arXiv preprint arXiv:2211.17115*, 2022.

[7] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus. Mathématique*, 350(5-6):313–318, 2012.

[8] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis. In *Forty-First International Conference on Machine Learning*, June 2024.

[9] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

[10] Daniel Garibi, Or Patashnik, Andrey Voynov, Hadar Averbuch-Elor, and Daniel Cohen-Or. ReNoise: Real Image Inversion Through Iterative Noising. *arXiv preprint arXiv:2403.14602*, 2024.

[11] Ligong Han, Yinxiao Li, Han Zhang, Peyman Milanfar, Dimitris Metaxas, and Feng Yang. Svdiff: Compact parameter space for diffusion fine-tuning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7323–7334, 2023.

[12] Ligong Han, Song Wen, Qi Chen, Zhixing Zhang, Kunpeng Song, Mengwei Ren, Ruijiang Gao, Anastasis Stathopoulos, Xiaoxiao He, Yuxiao Chen, Di Liu, Qilong Zhangli, Jindong Jiang, Zhaoyang Xia, Akash Srivastava, and Dimitris Metaxas. Improving Tuning-Free Real Image Editing with Proximal Guidance. *arXiv preprint arXiv:2306.05414*, 2023.

[13] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-Prompt Image Editing with Cross Attention Control. *arXiv preprint arXiv:2208.01626*, 2022.

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

[16] Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An Edit Friendly DDPM Noise Space: Inversion and Manipulations. *arXiv preprint arXiv:2304.06140*, 2024.

[17] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Direct Inversion: Boosting Diffusion-based Editing with 3 Lines of Code. *arXiv preprint arXiv:2310.01506*, 2023.

[18] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-Based Real Image Editing With Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023.

[19] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2426–2435, 2022.

[20] Jeongsol Kim, Yeobin Hong, Jonghyun Park, and Jong Chul Ye. FlowAlign: Trajectory-Regularized, Inversion-Free Flow-based Image Editing. *arXiv preprint arXiv:2505.23145*, 2025.

[21] Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. VIEScore: Towards Explainable Metrics for Conditional Image Synthesis Evaluation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12268–12290, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[22] Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. FlowEdit: Inversion-Free Text-Based Editing Using Pre-Trained Flow Models. *arXiv preprint arXiv:2412.08629*, 2025.

[23] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1931–1941, 2023.

[24] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

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

[26] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in neural information processing systems*, 34:18878–18890, 2021.

[27] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow, 2022.

[28] Qi Mao, Lan Chen, Yuchao Gu, Mike Zheng Shou, and Ming-Hsuan Yang. Tuning-Free Image Editing with Fidelity and Editability via Unified Latent Diffusion Model. *arXiv preprint arXiv:2504.05594*, April 2025.

[29] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. In *International Conference on Learning Representations*, 2021.

[30] Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt Inversion: Fast Image Inversion for Editing with Text-guided Diffusion Models. *arXiv preprint arXiv:2305.16807*, 2024.

[31] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. NULL-Text Inversion for Editing Real Images Using Guided Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023.

[32] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 4296–4304, 2024.

[33] Toan Nguyen, Kien Do, Duc Kieu, and Thin Nguyen. H-Edit: Effective and Flexible Diffusion-Based Editing via Doob's h-Transform, 2025.

[34] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot Image-to-Image Translation. In *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH '23, pages 1–11, New York, NY, USA, 2023. Association for Computing Machinery.

[35] Maitreya Patel, Song Wen, Dimitris N. Metaxas, and Yezhou Yang. Steering Rectified Flow Models in the Vector Field for Controlled Image Generation. *arXiv preprint arXiv:2412.00100*, 2024.

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

[37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[38] Litu Rout, Yujia Chen, Nataniel Ruiz, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Semantic Image Inversion and Editing using Rectified Stochastic Differential Equations. *arXiv preprint arXiv:2410.10792*, 2024.

[39] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023.

[40] Joonghyuk Shin, Alchan Hwang, Yujin Kim, Daneul Kim, and Jaesik Park. Exploring Multimodal Diffusion Transformers for Enhanced Prompt-based Image Editing. *arXiv preprint arXiv:2508.07519*, 2025.

[41] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. *arXiv preprint arXiv:2010.02502*, 2020.

[42] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[43] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[44] Łukasz Staniszewski, Łukasz Kuciński, and Kamil Deja. There and Back Again: On the relation between Noise and Image Inversions in Diffusion Models. *arXiv preprint arXiv:2410.23530*, 2026.

[45] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2023.

[46] Andrey Voynov, Qinghao Chu, Daniel Cohen-Or, and Kfir Aberman. p+: Extended textual conditioning in text-to-image generation. *arXiv preprint arXiv:2303.09522*, 2023.

[47] Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: Exact diffusion inversion via coupled transformations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22532–22541, 2023.

[48] Jiangshan Wang, Junfu Pu, Zhongang Qi, Jiayi Guo, Yue Ma, Nisha Huang, Yuxin Chen, Xiu Li, and Ying Shan. Taming Rectified Flow for Inversion and Editing. *arXiv preprint arXiv:2411.04746*, 2024.

[49] Siyuan Wang, Xi Yang, Zihao Zhou, Huiru Shao, Rui Zhang, Qiufeng Wang, Guangliang Cheng, and Kaizhu Huang. Rethinking Real Image Editing: Unleashing Diverse Editing Operators via Multi-Objective Optimization. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4150–4159, March 2026.

[50] Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-Free Image Editing with Natural Language. *arXiv preprint arXiv:2312.04965*, 2023.

[51] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. FreeDoM: Training-Free Energy-Guided Conditional Diffusion Model. *arXiv preprint arXiv:2303.09833*, 2023.

[52] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient Surgery for Multi-Task Learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 5824–5836. Curran Associates, Inc., 2020.

[53] Tianrui Zhu, Shiyi Zhang, Jiawei Shao, and Yansong Tang. KV-Edit: Training-Free Image Editing for Precise Background Preservation. *arXiv preprint arXiv:2502.17363*, 2025.

A Concepts in multi-objective optimization

Definition A.1. (Pareto Dominance.) For any two vectors $\boldsymbol{r}^1, \boldsymbol{r}^2 \in \mathcal{O}$, we denote $\boldsymbol{r}^1 \preceq \boldsymbol{r}^2$ if $\boldsymbol{r}^2 - \boldsymbol{r}^1 \in \mathbb{R}^n_+$, which implies $r^1_i \leq r^2_i, \forall i \in [n]$. If there further exists at least one $j \in [n]$ s.t. $r^1_j < r^2_j$, we have $\boldsymbol{r}^1 \prec \boldsymbol{r}^2$. Then, we say a solution $x_1 \in \mathcal{X}$ dominates another solution $x_2 \in \mathcal{X}$ iff $\boldsymbol{r}(x_1) \prec \boldsymbol{r}(x_2)$. Note if $\boldsymbol{r}(x_1) \not\prec \boldsymbol{r}(x_2)$, x_1 does not dominate x_2.

Definition A.2. (Pareto Optimal.) A solution $x^* \in \mathcal{X}$ is Pareto optimal if it is not dominated by any other solution, *i.e.*, $\boldsymbol{r}(x) \not\prec \boldsymbol{r}(x^*), \forall x \in \mathcal{X} - \{x^*\}$.

Definition A.3. (Pareto Set and Pareto Front.) The set of all the Pareto optimal solutions is given by $\mathcal{P} \stackrel{\text{def}}{=} \text{Pareto}\{x^* : \forall x \in \mathcal{X} - \{x^*\}, \boldsymbol{r}(x) \not\prec \boldsymbol{r}(x^*)\}$. Then, the corresponding set of multi-objective values of the Pareto set is called the Pareto front, denoted by $\mathcal{F} \stackrel{\text{def}}{=} \{\boldsymbol{r}(x^*), \forall x^* \in \mathcal{P}\}$.

B Algorithm

Algorithm 1 Constrained Look-Ahead Reward-guided Edit

Input: Source latent x^{src}, initial noisy latent x_T^{gen}, source condition c^{src}, edit condition c^{edit}, timesteps $\{t_k\}_{k=0}^{T}$, inner steps p, edit step size γ, fidelity threshold ε, projection step size η, max projection steps J

Output: Edited image $x^{\text{edit}} = x_0^{\text{gen}}$

for $k = T, T-1, \ldots, 1$ **do**
  $x \leftarrow x_k^{\text{gen}} + (t_{k-1} - t_k)\, v_\theta(x_k^{\text{gen}},\, t_k,\, c^{\text{edit}}), t \leftarrow t_{k-1}$
  for $i = 0, 1, \ldots, p-1$ **do**
    // **Version 1 (Edit-first):**
    $x_{\text{trial}} \leftarrow x + \gamma \nabla_x r_{\text{edit}}(x, t)$
    // **Version 2 (Look-ahead):**
    $x_{\text{edit}}(x) \leftarrow x + \alpha\, \nabla_x r_{\text{edit}}(x, t)$
    $x_{\text{fid}}(x) \leftarrow x + \beta\, \nabla_x r_{\text{fid}}(x, t)$
    $\mathcal{L}_{\text{LA}}(x) \leftarrow r_{\text{edit}}(x_{\text{edit}}(x), t) + \lambda\, r_{\text{fid}}(x_{\text{fid}}(x), t))$
    $x_{\text{trial}} \leftarrow x + \gamma\, \nabla_x \mathcal{L}_{\text{LA}}(x)$
    if $r_{\text{fid}}(x_{\text{trial}}, t) \geq \varepsilon$ **then**
      $x \leftarrow x_{\text{trial}}$
    else
      $x \leftarrow x_{\text{trial}}$
      for $j = 1, \ldots, J$ **do**
        if $r_{\text{fid}}(x, t) \geq \varepsilon$ **then**
          break
        end if
        $x \leftarrow x + \eta\, \nabla_x r_{\text{fid}}(x, t)$
      end for
    end if
  end for
  $x_{k-1}^{\text{gen}} \leftarrow x$
end for
return $x^{\text{edit}} \leftarrow x_0^{\text{gen}}$

C Edit reward options

We instantiate the edit reward $r_{\text{edit}}(x, t)$ using CLIP-based similarity scores computed on the predicted clean image:

$$\hat{x}_0(x, t, c^{\text{edit}}).$$

Let ϕ_{img} denotes the CLIP image encoder, and ϕ_{txt} denotes the CLIP text encoder. We use cosine similarity

$$\cos(a, b) := \frac{\langle a, b \rangle}{\|a\|_2 \|b\|_2}.$$

The edit reward can then be instantiated in several ways.

Text-image similarity. The simplest option directly measures the CLIP similarity between the predicted clean image and the target edit prompt:

$$r_{\text{clip_text}} = \cos\big(\phi_{\text{img}}(\hat{x}_0), \phi_{\text{txt}}(c^{\text{edit}})\big)\,. \tag{15}$$

This reward encourages the generated image to align with the target edit description.

Directional similarity. Following the common CLIP-directional formulation, we can instead compare the image-space change with the corresponding text-space change:

$$r_{\text{clip_directional}} = \cos\big(\phi_{\text{img}}(\hat{x}_0) - \phi_{\text{img}}(x^{\text{src}}), \phi_{\text{txt}}(c^{\text{edit}}) - \phi_{\text{txt}}(c^{\text{src}})\big)\,, \tag{16}$$

where x^{src} is the source image. This reward encourages the edit direction in image-embedding space to match the semantic change specified by the source and target prompts.

Contrastive similarity. We also consider a contrastive variant that increases similarity to the target prompt while decreasing similarity to the source prompt:

$$r_{\text{clip_contrastive}} = \cos\big(\phi_{\text{img}}(\hat{x}_0), \phi_{\text{txt}}(c^{\text{edit}})\big) - \cos(\phi_{\text{img}}(\hat{x}_0), \phi_{\text{txt}}(c^{\text{src}}))\,. \tag{17}$$

This reward is useful when direct target-prompt similarity alone is insufficient to encourage a clear semantic departure from the source prompt.

D Experimental Details

All experiments use FLUX.1-dev [24] as the base generative model. We run 28 denoising steps with guidance scale 3.5 for target condition. For each denoising step we apply $p = 3$ inner optimization iterations with edit step size $\gamma = 5.0$. The edit and fidelity probe distances are $\alpha = \beta = 5.0$, and all proposal gradients are ℓ_2 normalized before applying their respective step sizes. The fidelity projection uses step size $\eta = 5.0$ and a maximum of $J = 3$ correction steps per inner iteration. For Look-ahead mechanism, fidelity weight uses $\lambda_{\text{fid}} = 1$. The fidelity threshold is fixed at $\varepsilon = -0.1$ (latent-ℓ_2 metric) for main experiments. The edit reward is CLIP-contrastive. Forward initialization uses the noise level σ_T corresponding to the first denoising step of FLUX.1-dev.

E Additional Results on the FlowEdit Evaluation Set

FlowEdit [22] evaluates text-based real-image editing on a diverse collection of real images paired with source prompts and handcrafted target prompts. We follow this setting and report additional quantitative and qualitative results on all 273 examples of the released FlowEdit evaluation set. On this set, we use fidelity threshold $\varepsilon = -0.15$ and keep all other hyperparameters identical to the PIE-Bench experiments. We report CLIP-T for target-text alignment and LPIPS, DINO, and DreamSim for source preservation. We omit CLIP-I (source-to-edited image cosine similarity in CLIP space) from the table: as shown by [21], CLIP-I correlates poorly with human judgments of editing quality because CLIP embeddings operate at a coarse semantic level and are insensitive to pixel-level reconstruction errors, causing methods with poor fidelity to score spuriously high. Since this evaluation set does not provide edit masks, we do not report mask-based edit-region metrics.

Our method achieves higher target-text alignment and better source preservation than FlowEdit according to CLIP-T, LPIPS, DINO, and DreamSim. Figure 6 provides qualitative examples on the same evaluation set, spanning object replacement, color and attribute editing, and style transfer.

Table 4: Quantitative comparison on the FlowEdit evaluation set. LPIPS, DINO, and DreamSim measure source preservation, while CLIP-T measures target-text alignment. Best results are in **bold**.

Method	LPIPS ↓	DINO ↑	DreamSim ↓	CLIP-T ↑
FlowEdit [22]	0.1854	0.7402	0.2576	29.2862
Ours	**0.1573**	**0.8443**	**0.1144**	**29.6528**

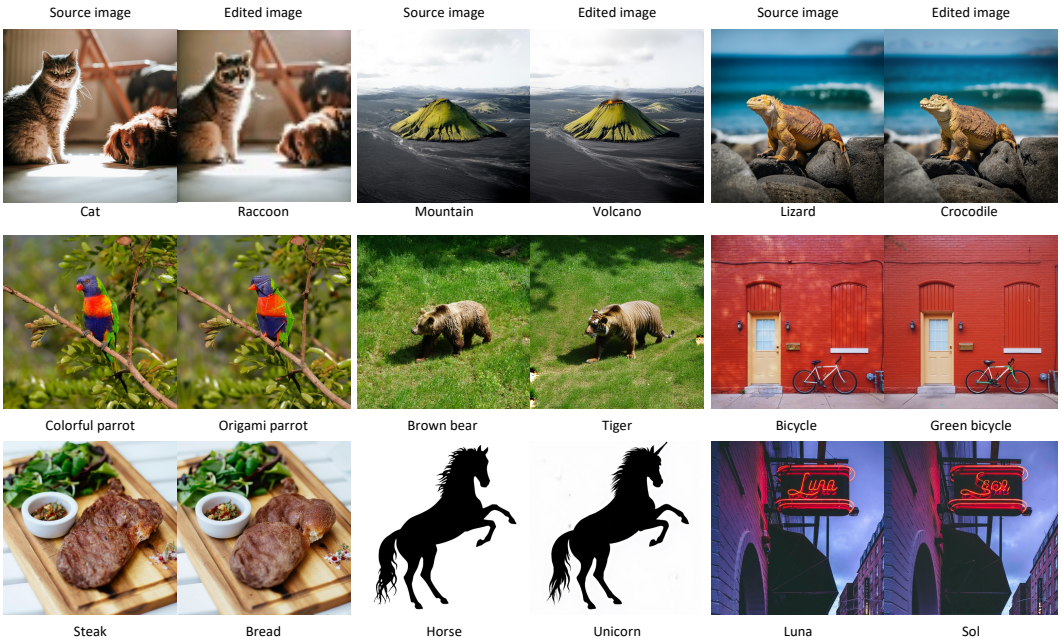

Figure 6: **Additional qualitative results on the FlowEdit evaluation set.** Our method executes diverse edit types, including object replacement, attribute modification, and style transfer, while preserving the structure of the original images.

F Additional Analyses

Scope of the trajectory objective. The Pareto formulation in Eq. (5) defines a solution set rather than a directly computable update. Both rewards are nonlinear functions of a high-dimensional latent, the pretrained flow model, the clean-state prediction, and, for image-space rewards, the decoder and reward network. Approximating the full Pareto set independently at every sampling step would require repeated optimization under many scalarization weights or constraints. Our ε-constraint selects a desired operating region, while p inner updates and at most J corrections provide a tractable local approximation rather than an exact solution of the full Pareto set.

Our use of a predicted clean state is a compatibility choice that permits training-free use of existing differentiable image-space rewards. The framework does not require this representation: a differentiable timestep-aware reward $r_t(x_t)$ can replace $R(\hat{x}_0(x_t, t))$ without changing the constrained proposal and correction structure. Such rewards may provide denser intermediate feedback, but require reward models calibrated to the latent space and noise schedule.

Post-intervention clean-endpoint prediction. The rewards in Eq. equation 13 are not evaluated directly on the probe latents. For objective $i \in \{\text{edit}, \text{fid}\}$, let

$$\phi_i(x) := \hat{x}_0(x, t, c_i), \qquad r_i(x, t) := R_i(\phi_i(x)), \qquad g_i(x) := \nabla_x r_i(x, t), \tag{18}$$

where c_i is the condition associated with the objective and R_i includes the differentiable image-space reward computation. Given probe distance $s_i \in \{\alpha, \beta\}$, we form

$$x_i = x + s_i g_i(x) \tag{19}$$

Table 5: Post-intervention clean-endpoint prediction versus explicitly applying one denoising transition before reward evaluation.

Method	Structure ↓	PSNR ↑	LPIPS ↓	MSE ↓	SSIM ↑	CLIP Similarity (Whole) ↑	CLIP Similarity (Edited) ↑	Runtime / image (s) ↓	Peak GPU memory (GiB) ↓
Look-ahead (Ours)	**0.0091**	27.2950	**0.0751**	0.0026	0.8476	**27.2370**	22.0130	**185.88**	**55.82**
Look-ahead with one-step denoising	0.0114	**27.3860**	0.0757	**0.0025**	**0.8510**	27.1912	**22.0686**	243.70	64.88

Table 6: Edit-first versus look-ahead on the full PIE-Bench using identical hyperparameters and unrestricted correction.

Update rule	Structure ↓	PSNR ↑	LPIPS ↓	MSE ↓	SSIM ↑	CLIP Similarity (Whole) ↑	CLIP Similarity (Edited) ↑
Edit-first	**0.0175**	**25.0160**	0.0965	**0.0045**	**0.8442**	**28.7452**	23.7910
Look-ahead	0.0176	25.0050	**0.0957**	**0.0045**	0.8433	28.6262	**23.7916**

and recompute $\phi_i(x_i)$ before evaluating the post-intervention reward. The probe therefore asks how a hypothetical intervention at the current timestep changes the clean endpoint predicted from the remaining generation dynamics. This differs from current-state scalarization and gradient-combination rules, which use objective gradients evaluated only at the unperturbed latent.

Exact differentiation through a probe gives

$$\nabla_x r_i(x + s_i g_i(x), t) = (I + s_i H_i(x))^\top g_i(x_i), \qquad H_i(x) := \nabla_x^2 r_i(x, t), \tag{20}$$

and therefore introduces a Hessian–vector correction. This curvature is not solely that of the image-space reward. Since $r_i = R_i \circ \phi_i$, the chain rule yields

$$H_i = J_{\phi_i}^\top H_{R_i} J_{\phi_i} + \sum_j [\nabla R_i(\phi_i)]_j \, H_{\phi_{i,j}}, \tag{21}$$

so the correction also captures the local geometry of the condition-specific clean-state prediction. Our default implementation stops gradients through $g_i(x)$ when constructing x_i. It consequently uses the first-order approximation $g_i(x_i)$ and avoids the Hessian–vector product in Eq. equation 20, while still recomputing each gradient after its objective-specific intervention. Thus, it is not equivalent to combining $g_{\text{edit}}(x)$ and $g_{\text{fid}}(x)$ at the current state. By evaluating gradients at these perturbed states rather than solely at the current latent, our approach anticipates how hypothetical updates affect the predicted clean endpoints and their associated rewards. This yields a locally non-myopic update without explicitly unrolling future denoising steps.

Relation to explicit denoising look-ahead. Let $\Phi^c_{t\to s}$ denote the exact conditional flow map from time t to s. Flow composition gives

$$\Phi^c_{t\to 0}(z) = \Phi^c_{t-h\to 0}\big(\Phi^c_{t\to t-h}(z)\big)\,. \tag{22}$$

Accordingly, predicting the clean endpoint directly from a probe and first advancing that probe by one denoising step lead to the same endpoint under the exact flow. With the practical velocity estimator and numerical solver, they need not be identical, but they remain close when the local flow is approximately straight, as encouraged by rectified-flow training. We evaluate both realizations on the fixed PIE-Bench ablation subset. Table 5 verifies this connection empirically: explicit one-step rollout gives comparable endpoint metrics with mixed metric-wise differences, while increasing runtime by approximately 31% and peak memory by approximately 16%. Recomputing the post-intervention clean endpoint therefore captures much of the useful information from one explicit future step at lower cost.

F.1 Update-Rule Ablation

We isolate the contribution of the look-ahead rule from the surrounding constrained framework. First, we evaluate edit-first and look-ahead on the full PIE-Bench using the same inversion-free initialization, rewards, constraints, and hyperparameters; only the proposal rule changes. Table 6 shows that the two rules reach nearly identical endpoint performance under unrestricted correction, with mixed wins across metrics. This indicates that the constrained framework accounts for most of the overall gain in Table 1.

We then control correction cost on the fixed PIE-Bench ablation subset. For each image, edit-first receives the corresponding projection budget used by look-ahead and stops correcting once that

Table 7: User study results on five-point Likert scales.

Method	Edit Compliance ↑	Source Fidelity ↑
DDIM Inversion	2.83	2.00
TABA	2.92	2.21
RF-Inversion	3.12	2.58
RF-Solver-Edit	3.33	3.29
FlowEdit	**3.46**	3.21
FlowAlign	3.04	2.83
Ours	3.33	**3.71**



Figure 7: Instructions and consent form shown to participants in the user study.

budget is exhausted. As shown in Table 3, look-ahead improves all source-preservation metrics and whole-image target alignment while retaining comparable edited-region alignment. Its main benefit is therefore to reach a favorable operating point with less reliance on corrective projections, rather than to shift the unrestricted endpoint frontier substantially.

F.2 User Study

To complement automatic metrics, we conducted a user study with eight adult participants recruited through Prolific. Participants rated anonymized outputs from seven methods on 20 non-sensitive PIE-Bench examples. Each participant evaluated 21 outputs using two five-point Likert scales: edit compliance and source fidelity. Table 7 shows that our method achieves the highest source-fidelity score while maintaining competitive edit compliance.

Participation was voluntary and restricted to adults, and participants provided electronic informed consent before beginning. They were compensated at £6 per hour in accordance with Prolific's minimum payment requirement. The study involved rating ordinary, non-sensitive images and posed no risks beyond routine web image viewing; these conditions were disclosed before consent. None of the conditions requiring prior ethical review under the Finnish National Board on Research Integrity (TENK) guidelines applied, so prior ethical review was not required.

Figures 7 and 8 show the participant instructions and an example evaluation screen.

F.3 Computational Cost

We benchmark our method and the baselines that we run locally on the same NVIDIA H200 GPU and report per-image runtime and peak allocated GPU memory in Table 2. We include ITOC [3], which performs full-horizon trajectory-control optimization, as context for another reward-gradient-based editor. Our method is substantially more expensive than methods with fixed inversion and generation procedures because it repeatedly evaluates both reward gradients and applies fidelity corrections during generation. This overhead is an important practical limitation.

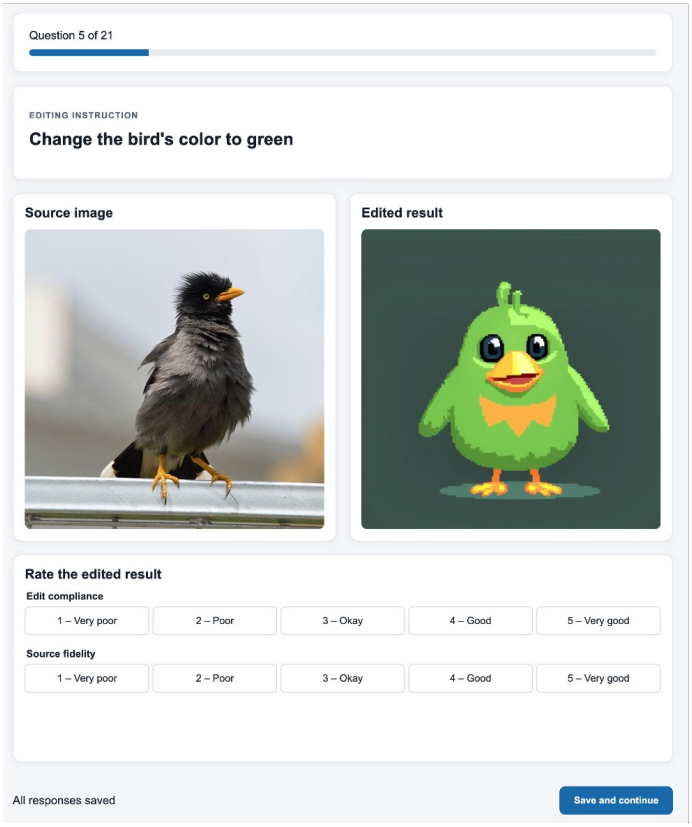

Figure 8: Example rating question from the user study. Method identities were hidden from participants.

Our default implementation uses a first-order stop-gradient approximation. Exact differentiation through the probe gradients introduces Hessian–vector products and requires 392.0 seconds and 129.76 GiB per image, compared with 185.88 seconds and 55.82 GiB for the first-order implementation, i.e., $2.11\times$ runtime and $2.32\times$ peak memory. Explicitly advancing each probe by one additional denoising step yields comparable endpoint metrics but increases runtime to 243.70 seconds and peak memory to 64.88 GiB. These results motivate post-intervention clean-endpoint prediction without exact second-order differentiation or explicit rollout.

We additionally consider a more efficient variant based on a linearized projection. Using the look-ahead objective $\mathcal{L}_{\mathrm{LA}}$ defined in Eq. equation 13, we denote its nominal latent increment by

$$\Delta x_{\mathrm{LA}} := \gamma \frac{\nabla_x \mathcal{L}_{\mathrm{LA}}(x)}{\|\nabla_x \mathcal{L}_{\mathrm{LA}}(x)\|_2}. \tag{23}$$

Here, Δx denotes an increment applied to the current latent x. We approximate the original fidelity constraint $r_{\mathrm{fid}}(x + \Delta x, t) \geq \varepsilon$ by its first-order expansion,

$$r_{\mathrm{fid}}(x,t) + \nabla_x r_{\mathrm{fid}}(x,t)^\top \Delta x \geq \varepsilon. \tag{24}$$

The projected update is the minimum Euclidean correction to the look-ahead increment:

$$\Delta x^\star = \arg\min_{\Delta x} \frac{1}{2}\|\Delta x - \Delta x_{\mathrm{LA}}\|_2^2 \quad \text{s.t.} \quad r_{\mathrm{fid}}(x,t) + \nabla_x r_{\mathrm{fid}}(x,t)^\top \Delta x \geq \varepsilon. \tag{25}$$

Its Karush–Kuhn–Tucker solution is

$$\Delta x^\star = \Delta x_{\mathrm{LA}} + \mu^\star \nabla_x r_{\mathrm{fid}}(x,t), \qquad \mu^\star = \left[\frac{\varepsilon - r_{\mathrm{fid}}(x,t) - \nabla_x r_{\mathrm{fid}}(x,t)^\top \Delta x_{\mathrm{LA}}}{\|\nabla_x r_{\mathrm{fid}}(x,t)\|_2^2}\right]_+, \tag{26}$$

where $\mu^\star$ is the non-negative Lagrange multiplier and $[z]_+ := \max(z, 0)$ denotes the positive-part operator. We then update the latent as $x \leftarrow x + \Delta x^\star$. Thus, the look-ahead increment is unchanged

Table 8: Sensitivity to inner updates p and maximum correction steps J. Dashes denote quantities not applicable to the corresponding sweep.

Sweep	Value	Structure ↓	PSNR ↑	LPIPS ↓	MSE ↓	SSIM ↑	CLIP Whole ↑	CLIP Edited ↑	Violations/image ↓	Updates/image
p	1	0.0126	**27.4450**	0.0795	0.0025	0.8454	25.9090	21.1890	–	28.00
	2	0.0101	27.4370	0.0770	0.0025	**0.8522**	25.7660	21.2480	–	56.00
	3	**0.0091**	27.2950	**0.0751**	0.0026	0.8476	**27.2370**	22.0130	–	84.00
	4	0.0103	27.4060	0.0752	**0.0024**	0.8508	27.1870	**22.1060**	–	112.00
J	1	0.0129	27.2300	0.0835	0.0026	0.8409	26.4750	21.1020	64.22	–
	2	0.0101	27.2550	0.0781	**0.0025**	0.8456	**27.9370**	21.6040	55.61	103.56
	3	**0.0091**	**27.2950**	**0.0751**	0.0026	0.8476	27.2370	22.0130	52.56	139.94
	5	0.0094	27.2680	0.0755	0.0026	**0.8529**	27.0150	**22.4090**	**48.67**	203.94

Table 9: Sensitivity to fidelity weight λ, edit-probe distance α, and fidelity-probe distance β.

Sweep	Value	Structure ↓	PSNR ↑	LPIPS ↓	MSE ↓	SSIM ↑	CLIP Whole ↑	CLIP Edited ↑
λ	0.25	0.0094	27.3820	0.0769	0.0025	0.8498	26.9050	21.9750
	0.5	0.0094	27.2030	0.0745	0.0026	0.8471	27.1840	21.7970
	1.0	**0.0091**	27.2950	0.0751	0.0026	0.8476	**27.2370**	22.0130
	2.0	0.0105	27.3680	0.0771	0.0025	**0.8519**	26.7390	**22.4200**
	4.0	0.0103	**27.5620**	**0.0710**	**0.0024**	0.8518	26.5090	21.2290
α	1.0	0.0095	27.2720	0.0766	0.0026	0.8508	26.8310	21.8640
	2.5	0.0094	27.3070	0.0762	0.0026	0.8493	27.0360	**22.6770**
	5.0	**0.0091**	27.2950	**0.0751**	0.0026	0.8476	27.2370	22.0130
	7.5	0.0096	27.2650	0.0799	0.0026	0.8492	27.3760	22.6180
	10.0	0.0097	**27.3380**	**0.0751**	**0.0025**	**0.8529**	**27.6090**	22.1660
β	1.0	0.0101	**27.4360**	0.0764	**0.0025**	**0.8532**	27.4960	21.7550
	2.5	0.0096	27.4050	0.0779	**0.0025**	0.8527	27.3590	**22.5680**
	5.0	**0.0091**	27.2950	**0.0751**	0.0026	0.8476	27.2370	22.0130
	7.5	0.0097	27.2680	0.0759	0.0026	0.8501	26.9560	21.5660
	10.0	0.0095	27.1570	0.0781	0.0026	0.8503	**28.3000**	22.0030

when it satisfies the local fidelity constraint and otherwise receives the smallest first-order feasibility correction. This closed-form projection replaces iterative trust-region corrections and reduces the runtime from 185.9 to 139.4 seconds per image, corresponding to a $1.33\times$ speedup.

F.4 Sensitivity and Design Ablations

Unless noted otherwise, these ablations use the fixed PIE-Bench ablation subset and the same evaluation protocol. They vary one component at a time while keeping all remaining settings fixed.

Inner iterations and bounded correction. Table 8 reports sensitivity to the number of inner updates p and maximum correction steps J. Increasing p from 1 to 3 strengthens editing and improves several preservation metrics, whereas $p = 4$ provides only mixed marginal changes at additional cost. Increasing J improves approximate feasibility and preservation up to $J = 3$; moving to $J = 5$ increases the average number of corrections by roughly 46% with only marginal metric changes. The finite loop is therefore a computation-bounded approximation: if the constraint remains violated after J corrections, sampling continues from the last corrected state rather than guaranteeing exact feasibility.

Look-ahead weighting and probe distances. The weight λ acts on objective-specific post-intervention rewards, while feasibility is controlled separately by ε. Table 9 shows that preservation metrics vary modestly over a $16\times$ range of λ. We use $\lambda = 1$ without claiming that it optimizes every metric. We use the shared-step convention $\alpha = \beta = \gamma = 5$ with normalized gradients. Varying either probe distance from 1 to 10 produces a stable preservation profile and no uniformly optimal setting. In contrast, direct scalarization controls feasibility only indirectly through reward scales; its empirical edit–fidelity curve is dominated by the constrained method, particularly under strict fidelity requirements.

Reward and initialization choices. Table 10 compares the edit rewards described in Appendix C and the two initialization strategies. Text-image similarity attains the highest target CLIP but remains

Table 10: Edit-reward and initialization ablations. Source/target CLIP and margin are reported only for the reward comparison.

Type	Variant	Structure ↓	PSNR ↑	LPIPS ↓	MSE ↓	SSIM ↑	Source CLIP	Target CLIP ↑	CLIP Edited ↑	Margin ↑
Edit reward	Text-image	0.0100	27.4210	0.0765	**0.0025**	**0.8518**	34.4100	**37.1330**	**26.8110**	2.7230
	Directional	0.0094	**27.5340**	**0.0749**	**0.0025**	0.8507	22.3930	27.2300	22.4150	4.8370
	Contrastive	**0.0091**	27.2950	0.0751	0.0026	0.8476	18.7140	27.2370	22.0130	**8.5240**
Initialization	Inversion-free	**0.0091**	**27.2950**	**0.0751**	**0.0026**	**0.8476**	–	27.2370	**22.0130**	–
	Inversion	0.0155	25.5560	0.0845	0.0038	0.8385	–	**27.5780**	21.9290	–

Table 11: Guidance-reward replacement on the PIE-Bench ablation subset.

Reward	Structure ↓	PSNR ↑	LPIPS ↓	MSE ↓	SSIM ↑	CLIP Whole ↑	CLIP Edited ↑
CLIP-contrastive	**0.0091**	**27.2950**	**0.0751**	0.00260	**0.8476**	**27.2370**	22.0130
ImageReward	0.0125	27.1130	0.0777	**0.00256**	0.8475	26.1243	21.9768
HPS v2.1	0.0124	26.9636	0.0776	0.00274	0.8471	27.1997	**22.4951**

close to the source prompt; contrastive similarity produces the largest target–source margin and best structure distance. Inversion-free initialization improves every preservation metric and edited-region alignment, with comparable whole-image target alignment.

Alternative differentiable guidance rewards. Because using CLIP for both optimization and evaluation can introduce circularity, we replace CLIP-contrastive guidance with ImageReward and HPS v2.1 while keeping the constrained procedure and all other hyperparameters fixed. Table 11 shows comparable preservation and edited-region CLIP alignment under both non-CLIPScore objectives. The framework is therefore not tied to CLIP-contrastive guidance.

Fidelity reward. A DINO-only constraint improves edited-region alignment slightly but severely degrades low-level source preservation, as semantic features are less sensitive to exact color, texture, and local geometry. Combining DINO with latent-ℓ_2 can improve individual fidelity metrics, but introduces another weight-dependent trade-off and does not uniformly dominate latent-ℓ_2. For example, weight 0.020 improves LPIPS from 0.0751 to 0.0736 but reduces edited-region CLIP from 22.0130 to 20.6478. We therefore retain latent-ℓ_2 as the default and report the full sensitivity in Table 12.

G Qualitative Effect of Fidelity Threshold

Figure 9 illustrates how the fidelity threshold ε controls the edit strength on two representative examples. As ε becomes more negative (looser constraint), the edit progressively strengthens: in the first case, the bird's texture gradually transitions toward a wooden appearance while the overall composition is retained; in the second case, the subject shifts continuously from a bunny toward a pig, with intermediate values producing a plausible blend between the two. At tight ε (e.g., $\varepsilon = -0.1$), the result closely preserves the source while making only subtle edits; at loose ε (e.g., $\varepsilon = -0.5$), the edit is strongly executed at the cost of greater source deviation. This monotonic and smooth behavior confirms that ε provides an continuous control over the edit–fidelity trade-off.

H Failure Cases and Limitations

Failure cases. Our method tends to complete edits through conservative latent updates, which improves consistency with the source image. However, when the requested edit requires a large semantic or style change, this conservative tendency leads to insufficient edit strength, producing results that remain overly similar to the source. Figure 10 illustrates two representative cases: changing raspberries to strawberries on cupcakes, and converting a black-and-white line drawing to a colorful and detailed image—both of which are under-edited by our method.

Table 12: Fidelity-objective ablation at threshold $\varepsilon = -0.1$. The weight applies to the DINO term in the combined objective.

Objective	Weight	Structure ↓	PSNR ↑	LPIPS ↓	MSE ↓	SSIM ↑	CLIP Whole ↑	CLIP Edited ↑
Latent-ℓ_2	0	**0.0091**	27.2950	0.0751	0.0026	0.8476	27.2370	22.0130
DINO	–	0.2106	9.6213	0.4456	0.1427	0.5122	26.8204	**22.7671**
Latent-ℓ_2 + DINO	0.005	0.0118	26.9860	0.0862	0.0028	0.8478	26.5615	21.2238
	0.020	0.0096	**27.4742**	**0.0736**	**0.0024**	**0.8548**	26.6459	20.6478
	0.050	0.0173	26.6272	0.0952	0.0036	0.8392	25.7457	21.4099
	0.100	0.0154	27.0727	0.0851	0.0029	0.8453	26.9130	21.3573
	0.200	0.0310	25.2286	0.1138	0.0061	0.8030	26.4817	21.5588
	0.300	0.0280	25.5062	0.1032	0.0043	0.8060	**27.5068**	22.0360

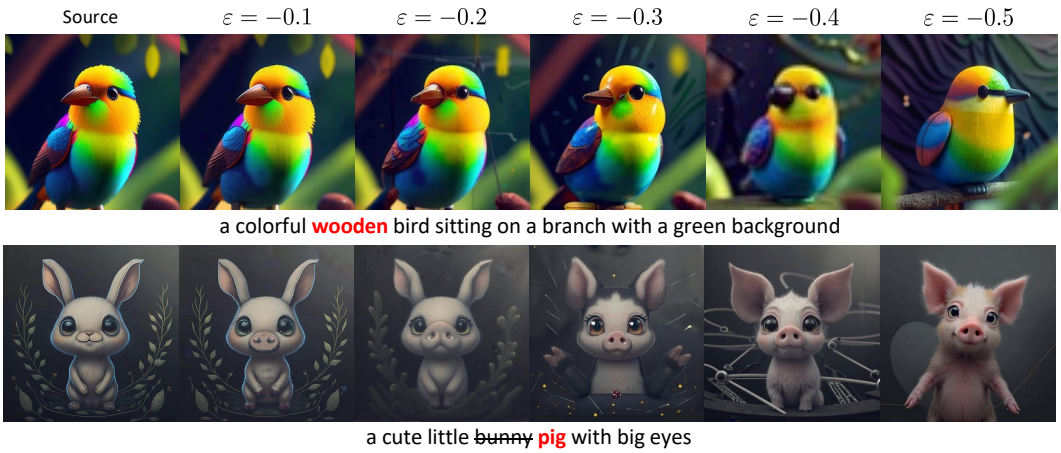

Figure 9: **Qualitative effect of varying ε.** Each row shows the source image followed by results under $\varepsilon \in \{-0.1, -0.2, -0.3, -0.4, -0.5\}$. Looser constraints produce stronger edits with greater source deviation.

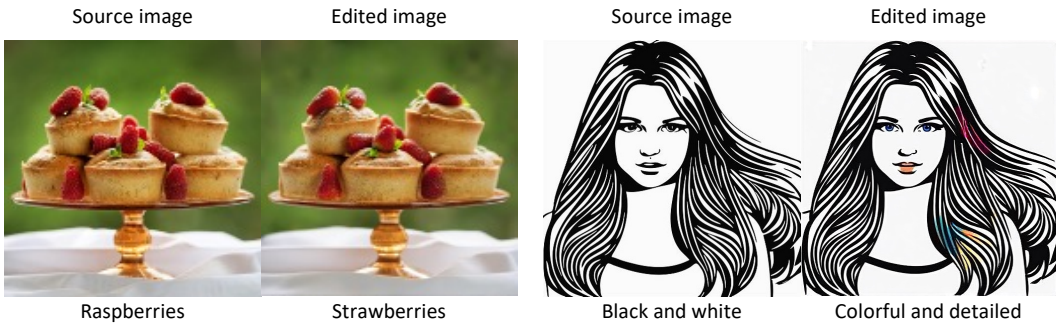

Figure 10: Failure cases. Our method produces results that are too similar to the source when the requested edit requires a large semantic or style change.

Limitations. First, the method introduces repeated reward-gradient evaluations and fidelity corrections at every generation step. It is therefore substantially slower and more memory intensive than editing methods with fixed inversion and generation procedures, as quantified in Appendix F.3; improving inference efficiency remains important future work. Second, a strict fidelity threshold can reject updates required for large geometric or semantic changes. Although relaxing ε provides an explicit control, the method cannot eliminate the fundamental tension between editability and source preservation and may under-edit demanding instructions. Third, the fidelity reward r_{fid} is a global measure; it does not enforce region-specific preservation, so edits can occasionally leak into areas that should remain unchanged. Finally, performance depends on reward quality: if the edit reward does not capture complex relational or compositional instructions, optimization may not converge to

the desired output. Alternative differentiable rewards are compatible with the framework, but their own biases remain.

I Broader Impacts

Our method improves controllability in text-guided image editing, which can benefit creative, educational, and accessibility applications. At the same time, like any generative editing technique, it could be misused to produce deceptive imagery. Our method relies on FLUX.1-dev [24]under a non-commercial research license that carries its own content restrictions, and standard generative model safeguards apply equally here.

J Code

The implementation code and execution instructions are available at `https://github.com/aalto-icl/ConstrainedLookaheadGuidance`.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the main contributions: constrained trajectory-level optimization and a look-ahead mechanism for reducing gradient conflict. All claims are empirically supported in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Appendix H and representative failure cases are shown there.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper provides necessary theoretical analysis in Section 3 and Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All hyperparameters are reported in Section 5.1 and Appendix D. The complete algorithm is described in Section 4, with implementation details in Appendix J.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The implementation code and execution instructions are publicly available at `https://github.com/aalto-icl/ConstrainedLookaheadGuidance`. PIE-Bench is publicly available from its original authors.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5.1 describes the dataset, baselines, and metrics. Full hyperparameters are in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Following standard practice in image editing benchmarks, we report mean metrics over the evaluation set in Section 5 without error bars.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources, per-sample runtime, and total wall-clock time are reported in Appendix F.3. No training is performed.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impacts are discussed in Appendix I.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We release no new model weights or datasets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets are explicitly credited and used under the appropriate license.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release the implementation code with installation and execution documentation. We do not release new datasets or model weights.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: Appendix F.2 reports the Prolific user-study protocol, complete participant instructions, and an example interface. Participation was voluntary and restricted to adults, participants provided electronic informed consent before beginning, and they were compensated at £6 per hour in accordance with Prolific's minimum payment requirement.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: The user study involved rating ordinary, non-sensitive images and posed no risks beyond routine web image viewing. These conditions and the voluntary nature of participation were disclosed before participants provided electronic consent. None of the conditions requiring prior ethical review under the Finnish National Board on Research Integrity (TENK) guidelines applied; therefore, prior ethical review was not required (`https://tenk.fi/en/ethical-review/ethical-review-human-sciences`).

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLM is only used for polishing the writing.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.