DECFLOWEDIT: SELF-LOCALIZED FLOW-BASED IMAGE EDITING VIA GUIDANCE DECOUPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow-based image editing (FlowEdit) enables inversion-free semantic changes through the difference between source and target velocities. In this paper, we observe that FlowEdit's default classifier-free guidance (CFG) configuration, with asymmetric source and target scales, causes substantial background leakage. Matching these guidance scales, for example by removing CFG, improves edit-relevant localization but severely degrades editability. To get the best of both worlds, we propose DecFlowEdit, which decouples the optimal guidance scales for localization and for editing in flow-based generative models. In particular, DecFlowEdit first extracts an edit-relevant prior by temporally aggregating velocity differences evaluated without CFG, and then uses this prior to reweight the original updates under default CFG. Our method remains training-free and inversion-free, requiring neither external spatial masks nor attention manipulation. Experiments on PIE-Bench across FLUX, SD3, and SD3.5 show that DecFlowEdit improves background preservation, reducing structure distance by approximately 61–73% and background LPIPS by 68–80% relative to FlowEdit at comparable editing fidelity.

1 INTRODUCTION

Text-guided image editing aims to realize a user-specified semantic change while preserving edit-irrelevant content. Achieving both objectives requires control over what changes and where those changes occur. Training-based methods learn this behavior from editing data (Brooks et al., 2023; Geng et al., 2023), whereas training-free methods exploit pretrained generative models without task-specific optimization. Despite their flexibility, training-free methods often face a trade-off: stronger semantic changes can introduce unintended modifications outside the edit region.

Many training-free editing methods follow an inversion-and-regeneration paradigm, first mapping the source image to a latent or noise representation and then regenerating it under the target condition (Su et al., 2023). However, preserving source content remains challenging, since these methods regenerate the whole image, including regions that should remain unchanged. Attention- and feature-control methods, including Prompt-to-Prompt (Hertz et al., 2023) and subsequent approaches (Tumanyan et al., 2023; Cao et al., 2023), improve background preservation by injecting source information into the generation process. Such interventions require access to internal model architectures and increase inference complexity. External spatial localization can also constrain the edit region, but it usually relies on additional localization signals (Couairon et al., 2023; Zhu et al., 2025) or external segmentation models (Li et al., 2023), which may be unavailable or may not align with the regions actually affected by the editing dynamics. Built on flow matching (Lipman et al., 2023), flow-based image editing (FlowEdit) (Kulikov et al., 2025) constructs paired stochastic source and target states and integrates their conditional velocity differences into a direct editing trajectory, without inversion or attention manipulation. Compared with inversion-and-regeneration (Su et al., 2023), this formulation improves source preservation, but its updates remain spatially global. Although its velocity differences are typically stronger in edit-relevant regions, residual differences in non-edit regions may accumulate across timesteps, causing background leakage (see Figure 1).

In this work, we identify the source and target classifier-free guidance (CFG) scales as a key factor governing this leakage. By default, FlowEdit uses asymmetric source and target guidance scales (default CFG) to achieve effective semantic editing. However, this asymmetry creates mismatched source and target velocity responses even in semantically unchanged regions, weakening their

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

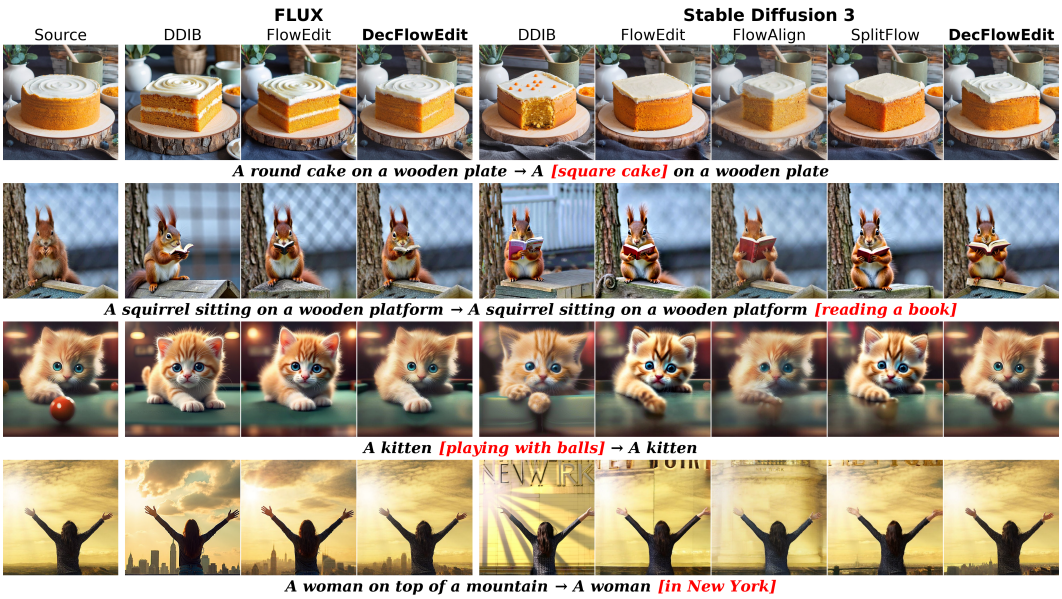

Figure 1: Qualitative comparison on PIE-Bench using FLUX (left) and SD3 (right). Text below each row summarizes the source-to-target prompt change, with edited phrases highlighted in red brackets. DecFlowEdit performs the intended edits while better preserving non-edited content and structure.

cancellation in the velocity difference. We observe that matching the guidance scales improves edit-relevant localization, especially when both scales are set to one, which removes CFG amplification. Without CFG (w/o CFG), velocity differences concentrate more closely on the intended edit region, but directly integrating them produces weak semantic changes (see Figure 2(a)). Thus, the guidance configuration that provides a localization signal differs from the one that enables effective editing.

Motivated by this observation, we propose **DecFlowEdit**, which decouples guidance for spatial localization from guidance for semantic editing. DecFlowEdit first extracts an edit-relevant spatial prior by temporally aggregating source–target velocity differences evaluated under w/o CFG. We then use the resulting soft prior to spatially reweight the FlowEdit updates under default CFG. This design separates the estimation of where to edit from the generation of the semantic change: the w/o CFG responses supply spatial localization, while the default CFG updates retain the strength needed for editing. Our main contributions are as follows: **1)** We identify a guidance-dependent trade-off between localization and editing in flow-based image editing: default CFG produces effective semantic changes but substantial background leakage, whereas w/o CFG improves localization at the cost of editability. **2)** We develop DecFlowEdit, which extracts an intrinsic spatial prior by temporally aggregating velocity differences under w/o CFG and uses it to reweight editing updates under default CFG. The method requires no training, inversion, external masks, or attention manipulation. **3)** We demonstrate consistent improvements in background and structure preservation on PIE-Bench across FLUX, SD3, and SD3.5 while maintaining comparable editing fidelity, supporting guidance decoupling as an effective approach to spatial control.

2 Related Work

Training-free image editing via inversion and attention manipulation. A line of training-free real-image editing methods follows the inversion-and-regeneration paradigm of Dual Diffusion Implicit Bridges (DDIB), where the source image is first mapped to a latent noise representation and then regenerated under the target condition (Su et al., 2023). Within this paradigm, Prompt-to-Prompt (Hertz et al., 2023) introduces cross-attention replacement to preserve spatial layout, inspiring subsequent attention- and feature-manipulation methods (Tumanyan et al., 2023; Cao et al., 2023; Wang et al., 2025a; Jiao et al., 2026). While effective in improving structure preservation, these methods have three practical drawbacks. First, their behavior is sensitive to the choice of

intervention layers and timesteps; inappropriate schedules or large source–target semantic gaps can introduce artifacts when preserved source patterns conflict with the intended target semantics. Second, they require overriding internal attention computations and maintaining source-branch representations during sampling, increasing implementation complexity and potentially incurring substantial inference overhead. Third, adapting these interventions to different backbones typically requires architecture-specific redesign.

FlowEdit and preservation-oriented flow editing. Built on flow matching models, FlowEdit constructs an inversion-free editing path from the difference between target- and source-conditioned velocity fields, yielding a more direct path between the source and target images and improved background preservation compared with inversion-based editing (Su et al., 2023; Lipman et al., 2023; Esser et al., 2024; Black Forest Labs, 2024; Kulikov et al., 2025). Recent follow-ups further improve content preservation and editing consistency through trajectory regularization (Kim et al., 2026), direct noise alignment (Xie et al., 2025), target-aware intermediate states (Wang et al., 2025b), or semantic flow decomposition (Yoon et al., 2025). However, under default CFG, FlowEdit's velocity differences can extend beyond the intended edit region, leaving non-edit regions susceptible to unintended changes. Our work is motivated by the observation that w/o CFG, despite producing weak semantic edits, yields more spatially localized velocity differences than default CFG. This observation suggests that the guidance requirements for spatial localization and effective editing differ. Accordingly, DecFlowEdit decouples guidance for these two purposes: we extract a spatial prior under w/o CFG and use it to constrain the updates under default CFG.

Spatial localization for background preservation. Spatial localization provides explicit control over the regions affected by image editing. Attention-based methods extract regions from internal attention maps (Cao et al., 2023; Patashnik et al., 2023), requiring access to the model's internal attention representations. Other approaches rely on user-provided masks or external segmentation models (Kirillov et al., 2023; Li et al., 2023; Zhu et al., 2025), introducing manual input or an additional model dependency. Prediction-based methods infer edit regions from source–target differences in noise predictions, as in DiffEdit (Couairon et al., 2023), or in velocity predictions, as in UniEdit-Flow (Jiao et al., 2026); because the contrast is taken on high-variance per-step predictions, the resulting masks are unstable and leak into non-edit regions. In contrast, our method obtains localization from FlowEdit's own velocity differences, which are more spatially localized w/o CFG. We aggregate these responses along a w/o CFG trajectory into a soft spatial prior. This temporal aggregation makes the prior more stable than one extracted from a single timestep. The resulting prior provides spatial control over editing without requiring attention manipulation, external segmentation models, or user-provided masks.

3 BACKGROUND

Conditional flow-based generation. Recent training-free editing methods are increasingly built upon flow matching models, which learn a continuous mapping from a data distribution $\mathbf{x}_0 \sim p_0$ to a prior $\mathbf{x}_1 \sim p_1$, typically the standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$ (Lipman et al., 2023). A linear interpolation $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1$, $t \in [0,1]$, defines the forward trajectory, and the trajectory follows $d\mathbf{x}_t = \mathbf{V}_\theta(\mathbf{x}_t, t \mid P)\, dt$, where $\mathbf{V}_\theta$ is the conditional velocity field and P denotes the textual condition. Flow matching trains the time-dependent conditional velocity field by minimizing $\mathcal{L}_\theta = \mathbb{E}_{t,\mathbf{x}_0,\mathbf{x}_1,P}[\|(\mathbf{x}_1 - \mathbf{x}_0) - \mathbf{V}_\theta(\mathbf{x}_t, t \mid P)\|^2]$.

Inversion-based editing using flow-based models. DDIB-style inversion-and-regeneration editing first maps a source image to a noise latent under the source prompt and then regenerates it under the target prompt (Su et al., 2023). Let P_{src} and P_{tar} denote the source and target prompts. The source-side inversion trajectory is obtained by integrating $d\mathbf{z}_t^{\mathrm{src}} = \mathbf{V}_\theta(\mathbf{z}_t^{\mathrm{src}}, t \mid P_{\mathrm{src}})\, dt$, starting from the source image $\mathbf{z}_0^{\mathrm{src}} := \mathbf{x}^{\mathrm{src}}$ and proceeding toward a noise latent $\mathbf{z}_1^{\mathrm{src}}$. Editing then integrates $d\mathbf{z}_t^{\mathrm{tar}} = \mathbf{V}_\theta(\mathbf{z}_t^{\mathrm{tar}}, t \mid P_{\mathrm{tar}})\, dt$ backward from the shared noise latent $\mathbf{z}_1^{\mathrm{tar}} = \mathbf{z}_1^{\mathrm{src}}$ to obtain the edited image $\mathbf{z}_0^{\mathrm{tar}}$. This shared latent provides only coarse structural control, and target-conditioned regeneration often introduces substantial changes to image details.

Inversion-free editing using velocity difference. FlowEdit constructs a direct editing path by explicitly pairing source and target states at each timestep, without first mapping the source image

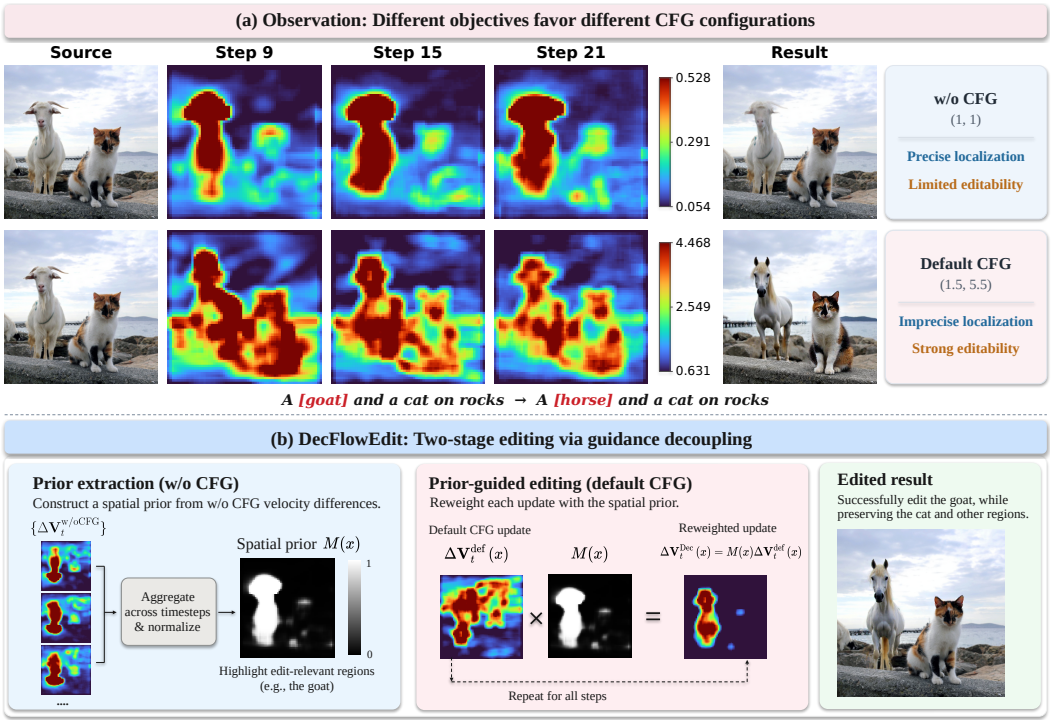

Figure 2: **Observation and overview of DecFlowEdit (FLUX).** (a) Velocity-difference magnitude maps and final editing results under w/o CFG and default CFG. Colorbars indicate magnitude scales. (b) Prior extraction under w/o CFG and prior-guided editing under default CFG.

to Gaussian noise (Kulikov et al., 2025). Following FlowEdit's time convention, the editing path starts from the source image $\mathbf{z}_1^{\text{edit}} := \mathbf{x}^{\text{src}}$ and is integrated backward toward the edited image $\mathbf{z}_0^{\text{edit}}$. At timestep t, given the current editing state $\mathbf{z}_t^{\text{edit}}$ and freshly sampled Gaussian noise $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the paired noisy states are $\mathbf{z}_t^{\text{src}} = (1-t)\mathbf{x}^{\text{src}} + t\boldsymbol{\epsilon}_t, \qquad \mathbf{z}_t^{\text{tar}} = \mathbf{z}_t^{\text{edit}} + \mathbf{z}_t^{\text{src}} - \mathbf{x}^{\text{src}}$. FlowEdit uses the difference between the two conditional velocity fields as the editing direction, $\Delta\mathbf{V}_t = \mathbf{V}_\theta(\mathbf{z}_t^{\text{tar}}, t \mid P_{\text{tar}}) - \mathbf{V}_\theta(\mathbf{z}_t^{\text{src}}, t \mid P_{\text{src}})$, and updates the editing trajectory by integrating $d\mathbf{z}_t^{\text{edit}} = \Delta\mathbf{V}_t\, dt$. Sharing the same noise within each timestep promotes similar source and target velocity responses in semantically unchanged regions, facilitating their cancellation in the velocity difference. This construction keeps editing anchored to the source image and improves content preservation compared with inversion-and-regeneration.

In practice, FlowEdit relies on strong, asymmetric source and target CFG scales for effective semantic editing. Asymmetric guidance can introduce a mismatch between the source and target responses even in semantically unchanged regions, weakening their cancellation. Stronger guidance can further amplify residual updates in these regions. Consequently, effective semantic editing can be accompanied by background leakage.

4 Method

We first examine how CFG affects spatial localization and semantic editing in FlowEdit (Section 4.1). We then describe the two stages of DecFlowEdit: *prior extraction*, which constructs a soft spatial prior from velocity differences under w/o CFG (Section 4.2), and *prior-guided editing*, which uses this prior to spatially constrain the updates under default CFG (Section 4.3).

4.1 Decoupling Localization from Editing

Let $\mathbf{g} = (g_{\text{src}}, g_{\text{tar}})$ denote the source and target CFG scales, and let $\Delta\mathbf{V}_t^{\mathbf{g}}$ be the corresponding FlowEdit velocity difference. We refer to $\mathbf{g}^{\text{w/oCFG}} = (1, 1)$, which applies no additional guidance

amplification, as *w/o CFG*. We use *default CFG* to denote FlowEdit's backbone-specific default source and target CFG scales.

For a CFG configuration $\mathbf{g}$, we visualize the spatial response of its velocity difference as

$$r_t^{\mathbf{g}}(x) = \left\|\Delta \mathbf{V}_t^{\mathbf{g}}(x)\right\|_2, \tag{1}$$

where x indexes a location in the latent spatial grid, and $\Delta \mathbf{V}_t^{\mathbf{g}}(x)$ denotes the channel vector of the velocity difference at that location, evaluated at the current paired source and target states. The norm is taken over channels.

Figure 2(a) shows the complementary behavior of the two CFG configurations. As indicated by the colorbars, FlowEdit under w/o CFG produces small-magnitude velocity differences that remain concentrated on the goat across steps 9, 15, and 21, leaving the final image almost unchanged. FlowEdit under default CFG produces velocity differences with much larger magnitudes and replaces the goat with a horse, but the updates also spread to the neighboring cat and cause an unintended change. These complementary behaviors motivate our two-stage editing framework (Figure 2(b)): the prior-extraction stage extracts a spatial prior under w/o CFG, and the prior-guided editing stage uses it to reweight updates under default CFG. This combines localized updates with effective semantic editing: the goat is replaced with a horse, while the cat and other regions are preserved.

4.2 Prior Extraction w/o CFG

In the prior-extraction stage, we compute velocity differences $\Delta \mathbf{V}_t^{\mathrm{w/oCFG}}(x)$ along an auxiliary FlowEdit trajectory $\mathbf{z}_t^{\mathrm{prior}}$ under w/o CFG. As Figure 2 shows, these responses already reveal the intended edit region at individual intermediate steps. To obtain a stable spatial prior, we aggregate velocity differences from the middle of the prior-extraction schedule, where spatial localization is relatively stable, before taking the norm:

$$E(x) = \left\| \sum_{t \in \mathcal{T}_{\mathrm{mid}}} \Delta \mathbf{V}_t^{\mathrm{w/oCFG}}(x) \right\|_2, \tag{2}$$

where $\mathcal{T}_{\mathrm{mid}} = \{t_k : 0.25T \le k \le 0.75T\}$, T is the total number of steps in the prior-extraction schedule, and t_k is the sampling time at step index k, with k starting from zero. We refer to $E(x)$ as the energy map. We then smooth the resulting spatial map using a 5×5 mean filter S, yielding $A(x) = S(E)(x)$. In practice, this stage can use a reduced sampling schedule to reduce computational cost. On FLUX, SD3, and SD3.5, a schedule with approximately one quarter as many sampling steps yields comparable editing and preservation performance (Section 5.2).

To reduce sensitivity to the response scale across samples, we normalize the smoothed map using its spatial percentiles:

$$\widetilde{E}(x) = \mathrm{clip}\left(\frac{A(x) - q_{10}}{\max\{q_{90} - q_{10}, \delta\}}, 0, 1\right), \qquad \delta = 10^{-8}, \tag{3}$$

where q_{10} and q_{90} are the 10th and 90th percentiles of all spatial values in A for the current sample.

The normalized response is converted into a soft spatial prior:

$$M(x) = \sigma\left(\frac{\widetilde{E}(x) - \tau}{s}\right), \qquad s > 0, \tag{4}$$

where σ is the sigmoid function, and τ and s denote the threshold and softness parameters.

4.3 Prior-guided Editing with Default CFG

In the prior-guided editing stage, we retain the spatial prior M and initialize a separate trajectory $\mathbf{z}_t^{\mathrm{edit}}$ from the source image and update it under default CFG. At each step, we construct the paired source and target states from $\mathbf{z}_t^{\mathrm{edit}}$ as in Section 3, evaluate their velocity difference $\Delta \mathbf{V}_t^{\mathrm{def}}$ under default CFG, and apply the spatial prior:

$$\Delta \mathbf{V}_t^{\mathrm{Dec}}(x) = M(x)\, \Delta \mathbf{V}_t^{\mathrm{def}}(x). \tag{5}$$

The scalar weight at each spatial location is broadcast across channels. Locations with large weights retain more of the default CFG update, while small weights suppress updates outside the edit region.

For consecutive sampling times $t_{k+1} \leq t_k$, the discrete update is

$$\mathbf{z}^{\text{edit}}_{t_{k+1}} = \mathbf{z}^{\text{edit}}_{t_k} + (t_{k+1} - t_k)\, M \odot \Delta \mathbf{V}^{\text{def}}_{t_k}, \qquad \mathbf{z}^{\text{edit}}_{t_0} = \mathbf{x}^{\text{src}}. \tag{6}$$

The velocity difference is re-evaluated along the spatially constrained trajectory at every step. We retain FlowEdit's paired-state construction and integration rule, inserting spatial weighting before each update. The same soft spatial prior is used throughout the prior-guided editing stage.

5 Experiments

5.1 Experimental Setup

Dataset and metrics. We evaluate our method on the Prompt-based Image Editing Benchmark (PIE-Bench) (Ju et al., 2023), a widely used benchmark for text-guided image editing. PIE-Bench contains 700 editing samples covering diverse editing types, and each sample provides a source image, source and target prompts, an editing instruction, and a ground-truth edit mask. Following the standard PIE-Bench evaluation protocol, we report (1) structure distance based on DINO features, where a lower value indicates better structure preservation, (2) background preservation based on PSNR, LPIPS, MSE, and SSIM on the non-edit region specified by the ground-truth mask, and (3) semantic editing fidelity based on CLIP similarity on the whole image and on the edited region.

Baselines. We compare our method with recent training-free flow-based image editing methods, including FlowCycle (Wang et al., 2025b), SplitFlow (Yoon et al., 2025), FlowAlign (Kim et al., 2026), UniEdit-Flow (Jiao et al., 2026), iRFDS (Yang et al., 2025), DRFS (Beaudouin et al., 2026), DNAEdit (Xie et al., 2025), and FTEdit (Xu et al., 2025). For a controlled comparison, we further compare our method with the corresponding FlowEdit (Kulikov et al., 2025) baseline on FLUX (Black Forest Labs, 2024), Stable Diffusion 3 (SD3), and Stable Diffusion 3.5 (SD3.5) (Esser et al., 2024).

Implementation details. In the prior-guided editing stage, we follow FlowEdit's default CFG, using 28 sampling steps with source/target guidance scales of $(1.5, 5.5)$ for FLUX, and 50 sampling steps with $(3.5, 13.5)$ for SD3/SD3.5. The prior-extraction stage uses FlowEdit under w/o CFG, i.e., source/target scales of $(1, 1)$. We construct the spatial prior as a soft map with threshold $\tau = 0.55$ and softness parameter $s = 0.08$. The main comparison uses the full sampling schedule for prior extraction (the same number of sampling steps as in the prior-guided editing stage). We also report a variant using a prior-extraction schedule with approximately one quarter as many sampling steps, while keeping the prior-guided editing stage fixed. More details are provided in Appendix C.

5.2 Main Results on PIE-Bench

Table 1 shows that DecFlowEdit yields consistent preservation gains: compared with raw FlowEdit, our method improves all background-preservation metrics and reduces structure distance by approximately 61–73% across FLUX, SD3, and SD3.5. These results show that the spatial prior extracted from w/o CFG responses can constrain unintended changes in the final edit. The preservation gains are accompanied by a 2–3% reduction in CLIP similarity. Across FLUX, SD3, and SD3.5, DecFlowEdit retains strong content preservation and competitive text alignment when the prior-extraction stage uses a sampling schedule with approximately one quarter as many steps (Table 1). This shorter schedule reduces the prior-extraction time while retaining useful spatial localization information.

Figure 1 further illustrates how this spatial control affects the final images. DecFlowEdit performs the intended semantic modifications while better preserving surrounding background textures and non-edited objects, reducing the unintended changes observed with raw FlowEdit and other baselines. The visible changes in the target region and the preserved surrounding content illustrate the division of roles in our two-stage design: the prior-extraction stage identifies where the edit should act, and the prior-guided editing stage performs the semantic edit under the resulting spatial constraint. We analyze how the prior threshold, softness, and update strength affect the editing–preservation trade-off in Appendix D (Figures 7–9).

Table 1: Quantitative comparison on PIE-Bench. Ours-short uses approximately one quarter as many sampling steps for prior extraction, while keeping the prior-guided editing stage unchanged. †: results reported in prior work.

Method	Backbone	Structure	Background Preservation				CLIP Similarity	
		Distance ↓	PSNR ↑	LPIPS$_{\times 10^3}$ ↓	MSE$_{\times 10^4}$ ↓	SSIM$_{\times 100}$ ↑	Whole ↑	Edited ↑
FlowEdit (Kulikov et al., 2025)	FLUX	28.99	21.72	104.79	98.32	83.87	25.88	22.71
Ours-short	FLUX	7.46	30.26	21.06	19.38	94.50	25.21	22.08
Ours	FLUX	7.86	30.54	20.90	19.23	94.54	25.25	22.15
FlowCycle† (Wang et al., 2025b)	SD3	13	26.83	63	30	88.6	25.48	22.46
SplitFlow† (Yoon et al., 2025)	SD3	14.55	25.22	68.53	44.96	87.54	26.23	23.01
iRFDS† (Yang et al., 2025)	SD3	62.72	19.61	186.39	179.76	74.59	24.54	21.67
FlowAlign (Kim et al., 2026)	SD3	36.70	24.02	65.02	56.91	86.55	26.65	22.70
UniEdit-Flow (Jiao et al., 2026)	SD3	21.31	25.22	94.80	46.06	86.71	26.45	22.87
FlowEdit (Kulikov et al., 2025)	SD3	26.39	22.27	100.15	85.71	83.93	27.51	23.92
Ours-short	SD3	10.35	27.93	32.69	28.94	91.62	26.66	23.35
Ours	SD3	10.01	28.16	31.34	27.85	91.75	26.71	23.37
SplitFlow† (Yoon et al., 2025)	SD3.5	11.68	27.12	52.93	30.61	89.76	26.29	22.89
DRFS (Beaudouin et al., 2026)	SD3.5	19.48	24.36	83.09	53.85	86.15	26.55	23.72
DNAEdit (Xie et al., 2025)	SD3.5	14.24	26.45	76.85	34.09	88.45	25.48	22.74
FTEdit (Xu et al., 2025)	SD3.5	18.40	25.23	92.36	41.59	88.77	23.96	21.17
FlowEdit (Kulikov et al., 2025)	SD3.5	22.55	23.27	88.78	69.66	85.57	27.62	23.95
Ours-short	SD3.5	9.10	28.68	29.71	25.06	91.99	26.90	23.47
Ours	SD3.5	8.90	28.78	28.79	24.59	92.03	26.86	23.35

5.3 CFG Affects Localization and Editing

To assess DecFlowEdit's division of roles between prior extraction and prior-guided editing, we examine how CFG affects localization and editing in FlowEdit. We evaluate localization accuracy using the IoU between masks extracted under different CFG settings and the ground-truth edit masks, and editing fidelity using edited-region CLIP similarity. We test w/o CFG, FlowEdit's default CFG, and symmetric CFG with guidance scale $g_{\text{src}} = g_{\text{tar}} > 1$ (see Appendix E.1 for detailed settings).

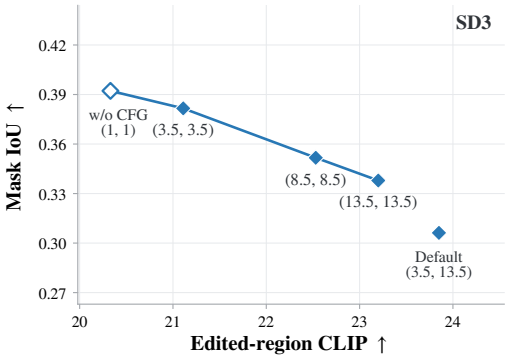

Figure 3: CFG trade-off on SD3. Without CFG $(1, 1)$ has the highest mask IoU; stronger symmetric CFG raises edited-region CLIP while reducing IoU. Default CFG $(3.5, 13.5)$ gives the highest CLIP and lowest IoU.

Figure 3 shows three trends. (i) Without CFG achieves the highest IoU, indicating that its responses align most closely with the intended edit region, but the lowest CLIP, consistent with incomplete realization of the desired semantic changes. (ii) Default CFG achieves the highest CLIP but the lowest IoU, indicating stronger semantic editing but greater leakage of updates into background regions. (iii) Symmetric CFG lies between these configurations: as guidance strength increases, IoU decreases while CLIP improves. We hypothesize that matching the source and target guidance strengths promotes cancellation of shared semantic components in their velocities. Increasing guidance strength, however, may amplify fluctuations in the branch velocities and leave larger residual updates outside the intended edit region. The first two trends agree with the qualitative results in Figures 2 and 4. These results support DecFlowEdit's two-stage design: the prior-extraction stage obtains localization information under w/o CFG, and the prior-guided editing stage uses it to constrain semantic editing under default CFG. We further compare priors extracted under w/o CFG and default CFG, keeping the editing stage fixed at default CFG. Priors extracted under w/o CFG reduce background LPIPS by approximately 39–50% relative to those extracted under default CFG, while edited-region CLIP differs by less than 0.13%. These results show that the localization advantage of w/o CFG translates into better background preservation in the final edit. Experimental settings and detailed results are provided in Appendix E.

Global amplification of w/o CFG updates. We additionally test whether amplifying w/o CFG updates can recover the editing performance of default CFG. At each step, we compute both updates



Figure 4: Qualitative comparison of ground-truth masks, extracted priors, and editing results from FlowEdit under w/o CFG and default CFG, together with norm-matched editing and DecFlowEdit, all using FLUX.

at the same paired source/target states with the same sampled noise, and multiply the w/o CFG update by a single global scalar to match the Frobenius norm of the default CFG update. The norm-matched outputs in column 7 of Figure 4 exhibit severe blur and loss of detail. Quantitatively, this approach yields lower edited-region CLIP and higher structure distance than default CFG across FLUX, SD3, and SD3.5 (Appendix E.4).

5.4 Understanding Guidance-dependent Localization

We analyze the localization advantage of w/o CFG through spatial and temporal differences between edit and non-edit regions. FlowEdit's shared-noise construction favors cancellation of common content between the source and target velocities (Appendix A.1). In the goat-to-horse example in Figure 2, both prompts describe the same neighboring cat, but default CFG uses different guidance strengths and can produce different velocities even for this unchanged object. Their difference can therefore contain unintended changes. Without CFG removes this guidance mismatch, favoring cancellation in non-edit regions.

Within-step spatial contrast. We compare unmasked FlowEdit trajectories on PIE-Bench, using ground-truth masks mapped to the latent grid to define the edit region Ω_E and its complement, the non-edit region Ω_N. At each update step, R_t is the ratio of the non-edit-region to edit-region mean velocity-difference magnitude. Averaging over update steps within each sample and then across samples gives 0.5175 under w/o CFG and 0.8007 under default CFG on FLUX. Thus, non-edit-region magnitudes are roughly half the edit-region magnitudes under w/o CFG, whereas default CFG makes the regions harder to distinguish. SD3 and SD3.5 show the same trend (Appendix A.2).

Temporal directional contrast. Beyond the within-step spatial contrast, we examine how velocity differences in edit and non-edit regions combine during temporal aggregation in Equation (2). More consistent directions favor accumulation in the vector sum. We therefore flatten each region's velocity difference into $\mathbf{u}_t^{\Omega} = \mathrm{vec}(\Delta \mathbf{V}_t|_{\Omega})$ and measure the mean cosine similarity over all distinct timestep

Table 2: Localization quality and downstream editing performance of different spatial priors on PIE-Bench. All metrics use the same 490 local-edit tasks (Appendix B). IoU and CLIP (edited region) are higher-better; LPIPS (background, $\times 10^3$) and Structure (distance) are lower-better.

Source	Model for prior-extraction	IoU	CLIP	LPIPS	Structure
DiffEdit-Single	FLUX	0.3392	20.860	30.119	8.351
DiffEdit-Multi	FLUX	0.3678	21.075	36.842	10.849
MasaCtrl	FLUX	0.3288	21.392	58.935	16.430
MasaCtrl	SD1.4	0.3087	21.071	55.702	14.903
Prompt-to-Prompt	SD1.4	0.3436	21.588	69.940	20.707
Auto VLM→SAM3	Qwen3-VL-8B + SAM3	0.5493	20.857	21.286	8.620
Ours	FLUX	0.4195	21.290	21.190	7.978

pairs in $\mathcal{T}_{\text{mid}} = \{t_1, \ldots, t_m\}$, where $m = |\mathcal{T}_{\text{mid}}|$ is the number of selected timesteps:

$$\text{Cos}_\Omega = \frac{2}{m(m-1)} \sum_{i<j} \frac{\langle \mathbf{u}_{t_i}^\Omega, \mathbf{u}_{t_j}^\Omega \rangle}{\|\mathbf{u}_{t_i}^\Omega\|_2 \, \|\mathbf{u}_{t_j}^\Omega\|_2}. \tag{7}$$

Averaged over PIE-Bench editing tasks performed with FLUX, the edit-region Cos exceeds the non-edit-region Cos by a larger margin under w/o CFG than under default CFG (0.12740 vs. 0.04984). Together with the cancellation analysis in Appendix A.3, this indicates that temporal aggregation under w/o CFG more selectively retains edit-region velocity differences while attenuating residuals in non-edit regions.

5.5 Comparison of Spatial Priors

To evaluate the spatial constraints provided by our prior-extraction stage, we compare our prior with localization signals from other common diffusion-based editing pipelines (Couairon et al., 2023; Cao et al., 2023; Hertz et al., 2023) and an external segmentation pipeline. The external pipeline uses Qwen3-VL-8B (Bai et al., 2025) as a vision-language model (VLM) to generate a query from the source image and editing instruction, followed by SAM3 (Carion et al., 2025) segmentation. Table 2 reports both IoU against the ground-truth edit masks and downstream editing and preservation metrics. Evaluation details and the mask extraction procedures for all baselines are provided in Appendix B.

Comparison with internal localization signals. DecFlowEdit obtains spatial information from the editing process by aggregating velocity differences under w/o CFG in the prior-extraction stage. Compared with the other internal localization methods in Table 2, our prior achieves higher IoU, lower background LPIPS, and lower structure distance, while retaining competitive edited-region CLIP. These results show that temporally aggregated velocity differences provide useful localization cues as well as effective spatial constraints for the prior-guided editing stage.

Comparison with external segmentation. In the end-to-end comparison with Auto VLM→SAM3, DecFlowEdit achieves higher edited-region CLIP and lower structure distance at similar background LPIPS, despite lower mask IoU. Our prior is intended to capture regions affected by editing. As a qualitative illustration, in the rusty-bicycle example in Figure 4, it highlights metal components such as the frame and rims rather than the entire object.

6 Conclusion

We present DecFlowEdit, a training-free and inversion-free method for improving spatial control in flow-based image editing. Our analysis reveals that the guidance configuration plays different roles in localization and semantic editing: w/o CFG favors localization, whereas default CFG favors effective semantic editing. By exploiting this distinction, DecFlowEdit improves the preservation of edit-irrelevant content without modifying the underlying generative model or requiring additional supervision. Experiments across three flow-based backbones show that this simple separation leads to more reliable editing trajectories and a better balance between semantic fidelity and structure preservation. More broadly, our results suggest that the internal dynamics of flow-based editors can provide useful spatial signals for controlling edits.

AI USE STATEMENT

We used generative AI tools to assist with writing and language polishing, identifying and retrieving related literature, and research ideation and execution, including code implementation, debugging, and preparation of figures and tables. The authors take responsibility for the final manuscript and accompanying artifacts.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Gaspard Beaudouin, Minghan Li, Jaeyeon Kim, Sung-Hoon Yoon, and Mengyu Wang. Delta rectified flow sampling for text-to-image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18662–18672, June 2026.

Black Forest Labs. FLUX. https://github.com/black-forest-labs/flux, 2024.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023.

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 *ICCV*, 2023.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025. URL https://arxiv.org/abs/2511.16719.

Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. In *ICLR*, 2023.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Han Hu, Dong Chen, et al. Instructdiffusion: A generalist modeling interface for vision tasks. *arXiv:2309.03895*, 2023.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross-attention control. In *ICLR*, 2023.

Guanlong Jiao, Biqing Huang, Kuan-Chieh Jackson Wang, and Renjie Liao. Uniedit-flow: Unleashing inversion and editing in the era of flow models. In *ICLR*, 2026.

Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Direct inversion: Boosting diffusion-based editing with 3 lines of code. *arXiv:2310.01506*, 2023.

Jeongsol Kim, Yeobin Hong, Jonghyun Park, and Jong Chul Ye. Flowalign: Trajectory-regularized, inversion-free flow-based image editing. In *ICLR*, 2026. URL https://openreview.net/forum?id=nyttIJfwW7.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. In *ICCV*, 2025.

Pengzhi Li, Qinxuan Huang, Yikang Ding, and Zhiheng Li. Layerdiffusion: Layered controlled image editing with diffusion models. In *SIGGRAPH Asia*. 2023.

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

Or Patashnik, Daniel Garibi, Idan Azuri, Hadar Averbuch-Elor, and Daniel Cohen-Or. Localizing object-level shape variations with text-to-image diffusion models. In *ICCV*, 2023.

Xuan Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. Dual diffusion implicit bridges for image-to-image translation. In *ICLR*, 2023.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, 2023.

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. In *ICML*, 2025a.

Yanghao Wang, Zhen Wang, and Long Chen. Flowcycle: Pursuing cycle-consistent flows for text-based editing. *arXiv preprint arXiv:2510.20212*, 2025b. URL https://arxiv.org/abs/2510.20212v1.

Chenxi Xie, Minghan Li, Shuai Li, Yuhui Wu, Qiaosi Yi, and Lei Zhang. Dnaedit: Direct noise alignment for text-guided rectified flow editing. *arXiv preprint arXiv:2506.01430*, 2025.

Pengcheng Xu, Boyuan Jiang, Xiaobin Hu, Donghao Luo, Qingdong He, Jiangning Zhang, Chengjie Wang, Yunsheng Wu, Charles Ling, and Boyu Wang. Unveil inversion and invariance in flow transformer for versatile image editing. In *CVPR*, 2025.

Xiaofeng Yang, Cheng Chen, Xulei Yang, Fayao Liu, and Guosheng Lin. Text-to-image rectified flow as plug-and-play priors. In *ICLR*, 2025.

Sung-Hoon Yoon, Minghan Li, Gaspard Beaudouin, Congcong Wen, Muhammad Rafay Azhar, and Mengyu Wang. Splitflow: Flow decomposition for inversion-free text-to-image editing. In *NIPS*, 2025.

Tianrui Zhu, Shiyi Zhang, Jiawei Shao, and Yansong Tang. Kv-edit: Training-free image editing for precise background preservation. *arXiv:2502.17363*, 2025.

A Additional Analysis of Guidance-dependent Localization

This appendix supplements Section 5.4 with a conceptual explanation, spatial-contrast measurements, and temporal directional and cancellation analysis.

A.1 Intuition for Guidance-dependent Localization

FlowEdit evaluates source and target velocities at paired states that share the same sampled noise. When the editing state stays close to the source image in a non-edit region, the paired states are locally similar, allowing velocity components associated with common content to partially cancel in their difference. Shared semantics alone, however, do not ensure identical velocities: default CFG applies different source and target guidance strengths, which can leave residual updates even on unchanged objects. In Figure 2, both prompts preserve the cat beside the goat, yet the guidance mismatch can produce different velocities on the cat. Without CFG substantially reduces this mismatch, favoring cancellation in non-edit regions. The spatial and temporal measurements in Section 5.4 quantify differences between edit and non-edit regions under the two CFG settings.

For the goat-to-horse example in Figure 2, Figure 5 evaluates both prompt branches at the same noisy source state at Step 15 of the 28-step FLUX schedule. We take the ℓ_2 norm over channels and smooth each magnitude map with a 5×5 mean filter. The four source/target maps share the pooled 10th–90th-percentile range $[3.616, 7.156]$, while each velocity-difference map uses its own percentile range.

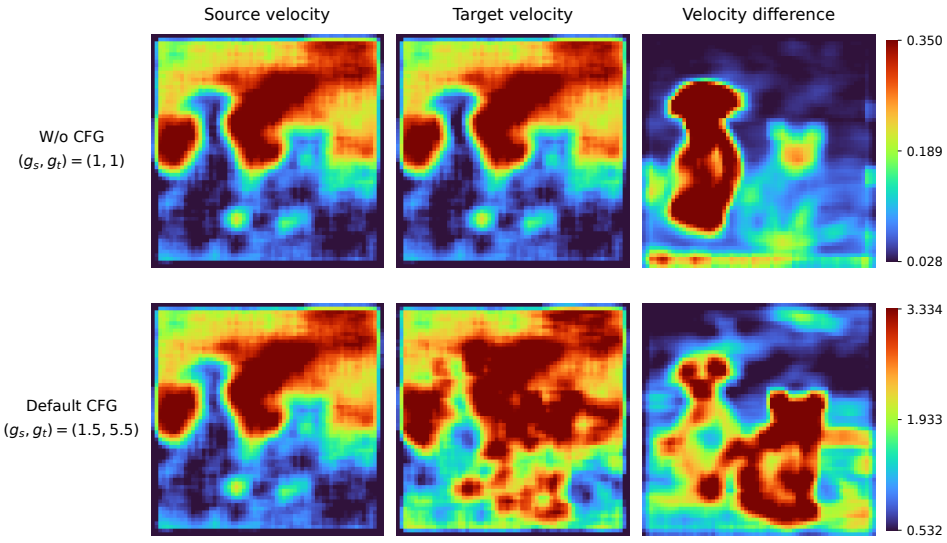

Figure 5: Magnitudes of the source velocity, target velocity, and their vector difference, evaluated at the same noisy source state at Step 15 under w/o CFG and default CFG (FLUX).

A.2 Within-step Spatial Contrast

We detail the spatial-contrast experiment in Section 5.4 on PIE-Bench tasks with nonempty edit and non-edit regions. Ground-truth edit masks are resized to the 64×64 latent grid by nearest-neighbor interpolation to define the edit region Ω_E; its complement is the non-edit region Ω_N. We compare unmasked FlowEdit trajectories under w/o CFG $(1, 1)$ and default CFG: $(1.5, 5.5)$ for FLUX and $(3.5, 13.5)$ for SD3 and SD3.5. Within each backbone, both configurations use the same inputs, sampling schedule, and paired noise draws, while their states evolve separately.

Let $r_t(x) = \|\Delta \mathbf{V}_t(x)\|_2$ denote the velocity-difference magnitude at location x, with the norm taken over channels. We compute the ratio of non-edit-region to edit-region mean magnitudes:

$$R_t = \frac{|\Omega_N|^{-1} \sum_{x \in \Omega_N} r_t(x)}{|\Omega_E|^{-1} \sum_{x \in \Omega_E} r_t(x)}. \tag{8}$$

For each sample, we average R_t over all actual update steps: indices 4–27 of FLUX's 28-step schedule and indices 17–49 of the 50-step SD3 and SD3.5 schedules, using zero-based indexing. We then average these per-sample values equally to obtain $\overline{R}$. The calculation uses raw velocity differences, without percentile normalization or temporal aggregation.

Switching from w/o CFG to default CFG increases $\overline{R}$ from 0.5175 to 0.8007 on FLUX, from 0.5985 to 0.8334 on SD3, and from 0.5903 to 0.8409 on SD3.5. The sample-averaged R_t is higher under default CFG at every actual update step: 24/24 on FLUX and 33/33 on each SD backbone. Thus, w/o CFG produces a clearer magnitude contrast between edit and non-edit regions across all three backbones. These compare the evolving trajectories; the per-step trend refers to sample means.

A.3 Temporal Directional Contrast

Section 5.4 finds a larger gap in temporal directional consistency between edit and non-edit regions under w/o CFG. Here we detail the cosine computation and examine whether this directional contrast is accompanied by stronger relative cancellation of velocity differences in non-edit regions during temporal aggregation.

We use the FLUX trajectories and region definitions described above. Following Equation (7), $\mathbf{u}_t^{\Omega} = \mathrm{vec}(\Delta \mathbf{V}_t|_{\Omega})$ concatenates all spatial positions and channels within region Ω. The intermediate window $\mathcal{T}_{\mathrm{mid}}$ contains indices 7–21 of the full 28-step schedule. Cos averages the cosine similarities over all distinct timestep pairs in this window. To measure cancellation when these velocity differences are aggregated as in Equation (2), we additionally compute the temporal cancellation ratio:

$$\mathrm{TCR}_{\Omega} = \frac{\left\| \sum_{t \in \mathcal{T}_{\mathrm{mid}}} \mathbf{u}_t^{\Omega} \right\|_2}{\sum_{t \in \mathcal{T}_{\mathrm{mid}}} \|\mathbf{u}_t^{\Omega}\|_2 + \delta}, \tag{9}$$

where $\delta = 10^{-8}$. The raw vectors are summed with equal weights, without multiplying by timestep sizes. Lower TCR indicates that a smaller fraction of the summed per-step magnitudes remains after vector addition, corresponding to stronger cancellation. Unlike Cos, TCR depends on both directions and magnitudes. Both metrics are computed per sample and then averaged equally across samples; Δ denotes the edit-region value minus the non-edit-region value.

Table 3: Temporal directional contrast and cancellation on FLUX. Cos and TCR use the same intermediate steps, indices 7–21. E and N denote the edit and non-edit regions; Δ denotes E minus N.

CFG	Cos			TCR		
	E	N	$\Delta \uparrow$	E	N	$\Delta \uparrow$
w/o CFG	0.53305	0.40565	**0.12740**	0.74017	0.65547	**0.08470**
default CFG	0.55259	0.50275	0.04984	0.75860	0.72725	0.03135

Table 3 shows larger gaps between edit and non-edit regions under w/o CFG than under default CFG in both Cos (0.12740 vs. 0.04984) and TCR (0.08470 vs. 0.03135). These larger gaps indicate stronger contrast between the two regions in directional consistency and in the fraction of cumulative velocity-difference magnitude retained after vector aggregation. This stronger contrast helps temporal aggregation distinguish edit regions from non-edit regions, supporting the use of w/o CFG for prior extraction.

B Mask Extraction Baselines

We describe the FLUX-based baselines, SD1.4 attention-control masks, and the external Auto VLM→SAM3 pipeline compared in Table 2.

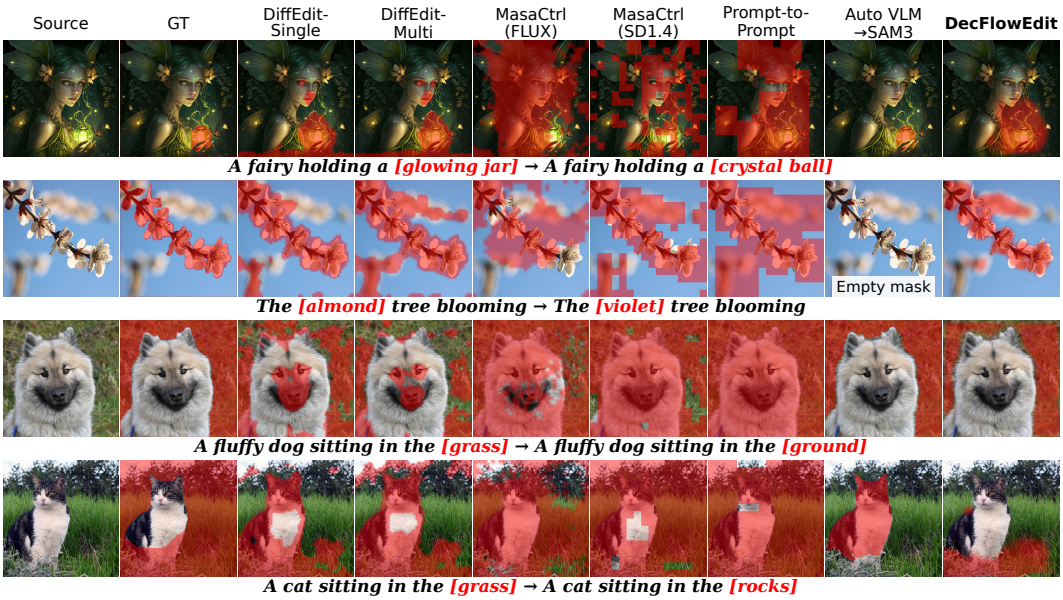

Figure 6: Additional qualitative comparison of different localization masks. In the second row, SAM3 returns no instance above the confidence threshold (0.5) for the VLM-generated query "almond tree", yielding an empty mask. GT: ground-truth edit mask.

DiffEdit-Single and DiffEdit-Multi. Both variants localize edits using source–target velocity disagreement on noisy versions of the source image. Given the source latent $\mathbf{x}^{\text{src}}$ and noise samples $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we construct $\mathbf{z}_{t_j}^{(i)} = (1-\sigma_j)\mathbf{x}^{\text{src}} + \sigma_j \boldsymbol{\epsilon}_i$ and compute

$$d^{(i,j)}(h,w) = \left\| \mathbf{V}_\theta(\mathbf{z}_{t_j}^{(i)}, t_j \mid P_{\text{tar}})(:,h,w) - \mathbf{V}_\theta(\mathbf{z}_{t_j}^{(i)}, t_j \mid P_{\text{src}})(:,h,w) \right\|_2 .$$

The localization map averages these magnitude maps over k timesteps and n noise samples:

$$\bar{d}(h,w) = \frac{1}{kn} \sum_{j=1}^{k} \sum_{i=1}^{n} d^{(i,j)}(h,w).$$

DiffEdit-Single uses $k = 1$, selecting a single noise level with strength 0.5 from a $T = 28$ schedule. DiffEdit-Multi uses $k = 5$ timesteps uniformly selected from $\sigma \in [0.2, 0.8]$. Both use $n = 10$ and source/target guidance scales of $(1, 1)$. We apply min–max normalization to $\bar{d}$ before binarization.

Our baselines adapt DiffEdit to FLUX, whereas the original work uses class-conditional latent diffusion and text-conditional Stable Diffusion models (Couairon et al., 2023). On 556 local-edit tasks, applying the original default threshold of 0.5 to our FLUX implementation yields low IoUs of 0.0702 and 0.1228 for DiffEdit-Single and DiffEdit-Multi, respectively. We therefore sweep the threshold and adopt $\tau = 0.15$ for both variants (IoUs of 0.3368 and 0.3615 on the same subset).

MasaCtrl (SD1.4 and FLUX). *SD1.4.* We use 512×512 images, source-image DDIM inversion, and 50-step DDIM sampling with guidance scale 7.5. PIE-Bench blend words identify the source and target control tokens. MasaCtrl (Cao et al., 2023) collects 16×16 cross-attention maps within each step, aggregates the selected-token maps over heads and layers, and applies min–max normalization to each branch. The attention store is reset after each step. Mutual self-attention control starts at step 4 and block 10 (zero-based). The target map is resized by nearest-neighbor interpolation to the current attention resolution and binarized using the official default threshold of 0.1. We export the target mask from the last control application at the final sampling step.

FLUX. The MasaCtrl-style mask is extracted from joint attention maps in the FLUX double-stream transformer blocks. We first locate the token indices corresponding to the edited word in the source prompt. During the forward pass, we extract the cross-attention submatrix from the source text tokens

to image patches:

$$\mathrm{Attn}_{\mathrm{edit}} = \mathrm{Attn}[:, :, \mathcal{I}_{\mathrm{token}}, S_{\mathrm{text}}:],$$

where $\mathcal{I}_{\mathrm{token}}$ denotes the token indices of the edited word and S_{text} is the text sequence length. We average the attention maps over tokens, heads, layers, and timesteps to obtain a spatial attention map. The map is min–max normalized to $[0, 1]$ and binarized at $\tau = 0.1$, using the same threshold as the official SD1.4 implementation. We use $T = 28$ and guidance scale 3.5.

These FLUX-based baselines represent different localization signals: DiffEdit-Single uses single-step source–target velocity disagreement, DiffEdit-Multi uses multi-step velocity-disagreement aggregation, and the MasaCtrl-style mask uses internal attention signals. In contrast, our spatial prior is derived directly from temporally aggregated FlowEdit velocity differences, making it better aligned with the editing operator used for subsequent reweighting.

Prompt-to-Prompt (SD1.4). We use 512×512 images, source-image DDIM inversion, and 50-step DDIM sampling with guidance scale 7.5. PIE-Bench blend words identify the source and target control tokens. We use the official attention-refinement controller (Hertz et al., 2023), with cross-attention and self-attention replacement fractions of 0.8 and 0.4. LocalBlend uses the 16×16 maps from `down_cross[2:4]` and `up_cross[:3]`, accumulated over all 50 sampling steps. It aggregates the selected-token maps over heads and layers, applies 3×3 max-pooling and nearest-neighbor interpolation, and divides each branch's map by its spatial maximum. The source and target maps are binarized using LocalBlend's official default threshold of 0.3 and combined by union. We export the hard mask used by LocalBlend at the final step.

Auto VLM→SAM3. Qwen3-VL-8B-Instruct (Bai et al., 2025) receives the source image, source and target descriptions, and editing instruction, and returns a single source-side object or part phrase. We use greedy decoding with at most 32 generated tokens. SAM3 (Carion et al., 2025) segments the source image using this phrase, with its image processor resizing the input to 1008×1008. We retain all instances with confidence above the official default threshold of 0.5, resize their mask logits to the source-image resolution, and apply a sigmoid. For IoU evaluation, we binarize each instance using the official pixel-probability threshold of 0.5 and take the union. For downstream editing, we instead take the pixel-wise maximum of the retained instance probabilities.

Evaluation protocol for the spatial-prior comparison. All metrics in Table 2 are evaluated on the same 490 local-edit tasks in PIE-Bench. Starting from its 700 tasks, we exclude 144 whose ground-truth masks cover the entire image and 66 for which the SD1.4 controllers cannot identify the required source–target token pair (46 with unparseable blend-word pairs and 20 with no exact token match). Mask extraction succeeds for both SD1.4 methods on the remaining $700 - 144 - 66 = 490$ tasks. The Model column identifies the model used to produce each prior; all priors guide FLUX editing under default CFG. IoU is measured against the ground-truth edit masks. Edited-region CLIP is scaled by 100, and background LPIPS and structure distance by 1000. For Auto VLM→SAM3, if the extracted mask is empty, as in the second row of Figure 6, we use an all-one weight map. The SD1.4 binary masks and SAM3 soft probability maps are saved at source-image resolution; SAM3 probabilities are stored as 8-bit grayscale values. For editing, the maps are converted to $[0, 1]$ and resized bilinearly to the FLUX latent resolution, without further thresholding, then used directly as the spatial update weights.

C Implementation Details

Backbones and environment. We evaluate DecFlowEdit on FLUX.1-dev, Stable Diffusion 3 (SD3), and Stable Diffusion 3.5-Medium (SD3.5), following the standard FlowEdit settings for each backbone. Experiments run on NVIDIA A100 GPUs under Ubuntu 24.04.3 and CUDA 13.0. FLUX.1-dev uses Python 3.8.20, PyTorch 2.4.1, and diffusers 0.30.1; SD3 and SD3.5 use Python 3.10.20, PyTorch 2.5.1+cu121, and diffusers 0.37.1. Table 4 reports compute resources and the runtime of a single FlowEdit editing stage.

Sampling schedules and CFG. The prior-guided editing stage uses 28 sampling steps for FLUX and 50 for SD3/SD3.5. Its default CFG configuration $\mathbf{g} = (g_{\mathrm{src}}, g_{\mathrm{tar}})$ is $(1.5, 5.5)$ for FLUX and $(3.5, 13.5)$ for SD3/SD3.5. The prior-extraction stage uses $\mathbf{g}^{\mathrm{w/oCFG}} = (1, 1)$. The full variant uses

Table 4: Compute resources and approximate per-image FlowEdit runtime on PIE-Bench.

Backbone	GPU	Peak Memory	FlowEdit Runtime
FLUX.1-dev	1×A100	33.31GB	8.1 sec/img
Stable Diffusion 3	1×A100	16.37GB	3.2 sec/img
Stable Diffusion 3.5-Medium	1×A100	17.08GB	3.9 sec/img

the same number of sampling steps for both stages, whereas Ours-short uses approximately one quarter as many steps for prior extraction while keeping the prior-guided editing stage unchanged. Relative to a single FlowEdit edit, the total sampling budget is approximately $2\times$ with the full prior-extraction schedule and $1.25\times$ with the short schedule. These ratios reflect the sampling-step counts of the two stages.

Spatial-prior settings. We follow the prior construction in Section 4.2, including the aggregation window $\mathcal{T}_{\text{mid}}$ defined by zero-based step indices, with threshold $\tau = 0.55$ and softness $s = 0.08$. For mask-quality evaluation, we binarize the soft spatial prior M at 0.5. The prior-guided editing stage uses the soft M throughout, following Section 4.3.

D Parameter Sensitivity and Editing–Preservation Trade-offs

We examine how update strength, the spatial-prior threshold τ, softness s, and the aggregation window affect localization and final editing. The first three controls change the amount or spatial distribution of the applied update; the window determines which responses form the prior. Unless specified otherwise, we use $\tau = 0.55$ and $s = 0.08$, and retain each backbone's default CFG in the prior-guided editing stage. These experiments characterize trade-offs rather than identify one setting that maximizes every metric.

D.1 Spatial Reweighting versus Uniform Scaling

We test whether the background-preservation gains in Table 1 can be explained solely by weaker editing. To assess the benefit of spatial selectivity, we compare spatial reweighting with uniform update scaling at comparable edited-region CLIP.

Control formulas. Let $\Delta\mathbf{V}_t^{\text{def}}(x)$ denote the velocity difference under default CFG and $M(x) \in [0, 1]$ the fixed soft spatial prior for an image. We compare

$$\Delta\mathbf{V}_t^{\text{Dec},\lambda}(x) = \big[\lambda + (1-\lambda)M(x)\big]\Delta\mathbf{V}_t^{\text{def}}(x), \tag{10}$$

$$\Delta\mathbf{V}_t^{\text{uniform},\lambda}(x) = \lambda\,\Delta\mathbf{V}_t^{\text{def}}(x). \tag{11}$$

Both controls recover raw FlowEdit at $\lambda = 1$. At $\lambda = 0$, spatial reweighting recovers the DecFlowEdit update in Equation (5), whereas uniform scaling supplies no editing update. Decreasing λ therefore strengthens suppression in both controls, but spatial reweighting targets low-prior locations while uniform scaling attenuates every location and channel equally. We compare the controls at comparable CLIP, not at equal λ. Each setting evaluates the velocity difference along its own trajectory.

Experimental settings and evaluation. Both controls use $\lambda \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ on PIE-Bench. We fix FLUX, default CFG $(1.5, 5.5)$, a 28-step schedule with 24 actual updates, and one fresh paired noise sample per step. The prior built from the aggregated response E (Equation (2)) uses $\tau = 0.55$ and $s = 0.08$. The sweeps share their raw-FlowEdit endpoint. Background LPIPS is evaluated on the 556 local tasks. At $\lambda = 0$ under uniform scaling, VAE encoding and decoding can still change pixels despite the absence of editing updates.

Results. At comparable edited-region CLIP, spatial reweighting achieves substantially lower background LPIPS than uniform scaling (Figure 7). Thus, spatial selectivity contributes to the preservation gains in Table 1 beyond the effect of uniformly reducing editing strength.

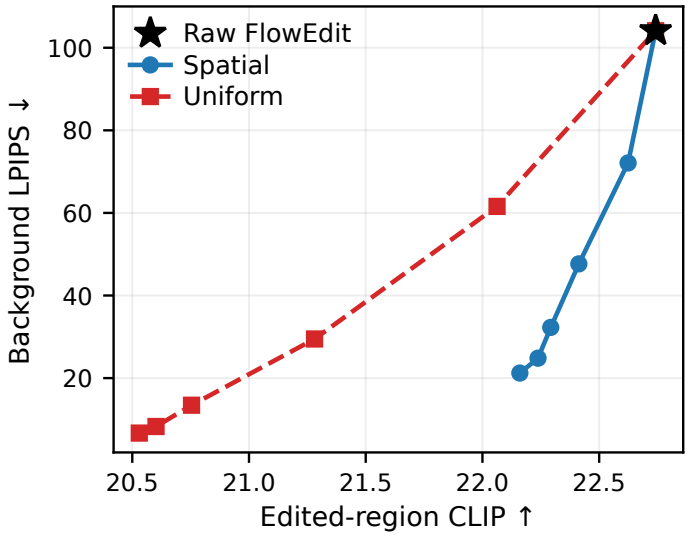

Figure 7: Spatial reweighting and uniform update scaling on FLUX. Each point is a measured control setting; the curves share the raw-FlowEdit endpoint. At comparable edited-region CLIP, spatial reweighting attains lower background LPIPS. Background LPIPS is scaled by 10^3.

D.2 Threshold: Localization and Downstream Editing

The threshold changes which locations receive large editing weights. We evaluate localization and downstream editing at $\tau \in \{0.30, 0.40, 0.50\}$ with fixed $s = 0.08$ across FLUX, SD3, and SD3.5. Both rows of Figure 8 additionally show $\tau = 0.55$ as a separate reference from an earlier run with the original (unquantized) priors.

The three-point sweep reconstructs the aggregated responses E from saved quantized response/mask images, rather than recovering the original floating-point maps. The underlying priors use E, 5×5 smoothing, and 10th/90th-percentile normalization, with recorded intermediate windows at indices 7–21 for FLUX and 17–38 for SD3/SD3.5 (zero-based). We evaluate localization by thresholding each soft prior at 0.5 and computing IoU on 556 local PIE-Bench tasks. For editing, we generate 700 outputs per setting; CLIP averages 700 tasks, and background LPIPS uses the 556 local tasks.

Among the displayed thresholds, $\tau = 0.40$ yields the highest mask IoU on all three backbones, whereas $\tau = 0.55$ favors more conservative spatial coverage. In the downstream sweep, increasing τ lowers both CLIP and background LPIPS: tighter spatial control improves preservation but limits the semantic change. The threshold with the highest binary-mask IoU therefore need not provide the desired final editing–preservation balance. We use $\tau = 0.55$ as a preservation-oriented operating point, not as a universal optimum.

D.3 Softness: Editing Changes despite an Unchanged Binary Mask

For $s > 0$, the sigmoid prior satisfies

$$M(x) = \sigma\left(\frac{\widetilde{E}(x) - \tau}{s}\right), \qquad M(x) > 0.5 \iff \widetilde{E}(x) > \tau. \tag{12}$$

Thus, changing s does not change the hard mask evaluated at threshold 0.5, although it changes the weights applied during editing. An offline FLUX check at $\tau = 0.55$ with $s \in \{0.04, 0.06, 0.08, 0.12, 0.16\}$ yields the same IoU, precision, and recall (0.4150, 0.6130, and 0.6293) for every setting.

To evaluate the effect on final images, we separately fix $\tau = 0.30$ and sweep $s \in \{0.04, 0.08, 0.16, 0.32\}$ on FLUX. We use the reconstructed priors and default CFG $(1.5, 5.5)$; the $s = 0.08$ outputs are reused from the threshold sweep. All plotted metrics average the same 556

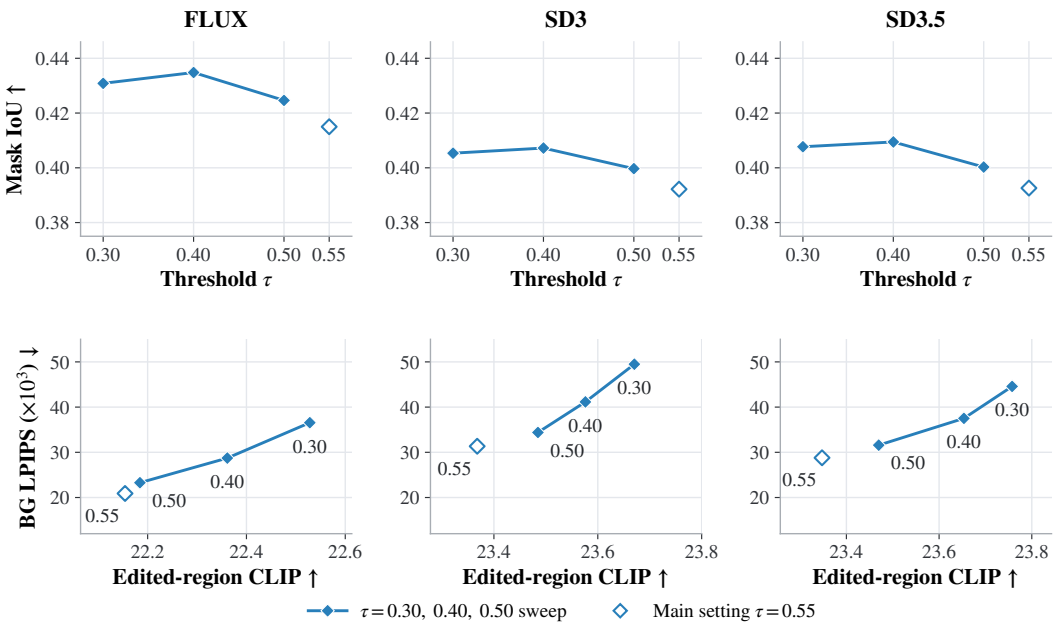

Figure 8: Threshold sensitivity across three backbones at fixed $s = 0.08$. Top: τ versus mask IoU. Bottom: edited-region CLIP versus background LPIPS, with points labeled by τ. Both rows show $0.30/0.40/0.50/0.55$: solid lines connect the three settings evaluated with reconstructed quantized priors, while unconnected hollow markers show the 0.55 reference from an earlier run with the original (unquantized) priors. IoU and background LPIPS use 556 local tasks; CLIP uses 700 tasks.

local tasks. Consequently, the CLIP values are not directly comparable to the 700-task averages in Figure 8.

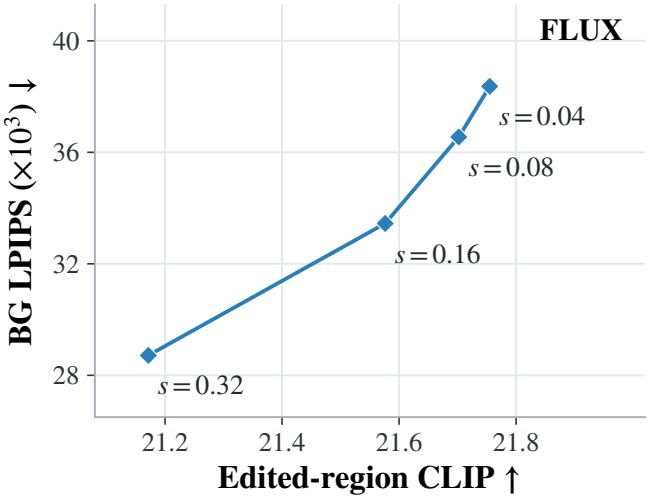

Figure 9: Softness sensitivity on FLUX at fixed $\tau = 0.30$. Point labels give s; both edited-region CLIP and background LPIPS use the same 556 local tasks. Increasing s in this sweep improves preservation while lowering CLIP, although the binary mask at threshold 0.5 is unchanged.

Figure 9 shows that s changes the editing–preservation trade-off even when hard-mask localization is invariant. Increasing s lowers both CLIP and background LPIPS in this sweep. The softness parameter must therefore be assessed through the final edits, rather than binary IoU alone.

D.4 Aggregation-window Sensitivity

We keep the full FLUX prior-extraction trajectory fixed (28 scheduled steps, with 24 actual updates at indices 4–27) and aggregate three subsets of its velocity differences: all available steps (4–27), an intermediate window (7–21), and a narrower intermediate window (11–17). We use $\tau = 0.55$, $s = 0.08$, and default CFG $(1.5, 5.5)$ for prior-guided editing. This changes the responses included in the prior, not the number of updates executed during prior extraction.

Table 5: Aggregation-window sensitivity on FLUX. Indices are zero-based; the prior-extraction trajectory executes all 24 updates for every row. IoU is evaluated on 556 local tasks using masks from the same trajectory. Edited CLIP uses 700 tasks and background LPIPS uses the 556 local tasks. †: downstream editing reuses an earlier run with the original (unquantized) priors; its mask differs from the intermediate-window mask used for the paired IoU comparison.

Window	Indices	Responses	IoU ↑	Edited CLIP ↑	BG LPIPS$_{\times 10^3}$ ↓
Wide	4–27	24	0.4028	22.0479	21.1075
Intermediate†	7–21	15	0.4096	22.1543	20.8951
Narrow	11–17	7	0.4118	22.0093	21.2234

The small differences in localization and editing performance across the tested windows (Table 5) support the stability of the spatial localization signals in w/o CFG velocity differences and their robustness to the aggregation-window choice.

D.5 Ground-truth-mask Diagnostic

We also constrain FLUX updates with the binary ground-truth edit mask, resized to latent resolution without additional dilation or smoothing. This diagnostic uses CFG $(1.5, 5.5)$ and a 28-step schedule with 24 actual updates.

Table 6: Ground-truth (GT) mask diagnostic on FLUX. Constraining edits with annotated regions improves preservation while also reducing CLIP similarity. Background SSIM, LPIPS, and MSE are scaled by 10^2, 10^3, and 10^4, respectively.

Method	Edited CLIP ↑	Whole CLIP ↑	BG SSIM ↑	BG LPIPS ↓	BG PSNR ↑	BG MSE ↓	Structure ↓
FlowEdit	22.71	25.88	83.87	104.79	21.72	98.32	28.99
FlowEdit + GT mask	22.41	25.27	96.23	7.76	36.44	4.46	9.75

Even with annotated edit regions, CLIP decreases from 22.71 to 22.41 (Table 6). A CLIP decrease alone therefore does not establish inaccurate automatic localization. The GT mask also defines the background evaluation region, giving this variant an oracle advantage; its preservation scores serve as diagnostic references rather than a fair baseline for automatic methods.

E CFG Configurations: Localization and Editing

This section supplements Section 5.3 with the experimental settings, additional backbone results, the prior-extraction CFG ablation, and the global norm-matching control.

E.1 Experimental Settings

For the direct-editing comparisons, we evaluate FlowEdit on the same 700 PIE-Bench tasks at 512×512 resolution, with seed 42 and one noise sample per step. Within each backbone, the configurations use the same input images and paired noise draws. Source and target branches share noise within a step, and fresh noise is sampled across steps. We compare w/o CFG $(1, 1)$, default CFG, and three symmetric settings $g_{\text{src}} = g_{\text{tar}} = g$ (Table 7). Images are generated directly without a spatial mask. The prior is subsequently extracted from each run's velocity differences to evaluate localization.

Table 7: CFG settings and sampling schedules. Guidance pairs are (source, target). All backbones also use w/o CFG $(1, 1)$. The final column distinguishes scheduled steps from actual updates.

Backbone	Default CFG	Symmetric g	Scheduled / actual steps
FLUX	$(1.5, 5.5)$	$1.5,\ 3.5,\ 5.5$	28 / 24
SD3	$(3.5, 13.5)$	$3.5,\ 8.5,\ 13.5$	50 / 33
SD3.5	$(3.5, 13.5)$	$3.5,\ 8.5,\ 13.5$	50 / 33

Prior construction and evaluation. We use the aggregated response E (Equation (2)), 5×5 smoothing, 10th/90th-percentile normalization, and the same sigmoid parameters $\tau = 0.55$ and $s = 0.08$. The recorded aggregation windows are zero-based indices 7–21 for FLUX and 17–38 for SD3/SD3.5. We binarize each soft prior at 0.5 and compare it with the ground-truth edit mask. IoU and background metrics use the 556 local tasks; structure distance is evaluated over the full image for those same local tasks. Displayed CLIP and SSIM are scaled by 10^2, LPIPS and structure distance by 10^3, and MSE by 10^4; PSNR is measured in dB.

E.2 Additional Results on FLUX and SD3.5

Figure 10 extends the SD3 comparison in Section 5.3. Both backbones exhibit the same trends: w/o CFG has the highest prior IoU but the lowest edited-region CLIP among the tested configurations, whereas default CFG has the highest CLIP and lowest IoU. Increasing symmetric guidance raises CLIP while reducing IoU.

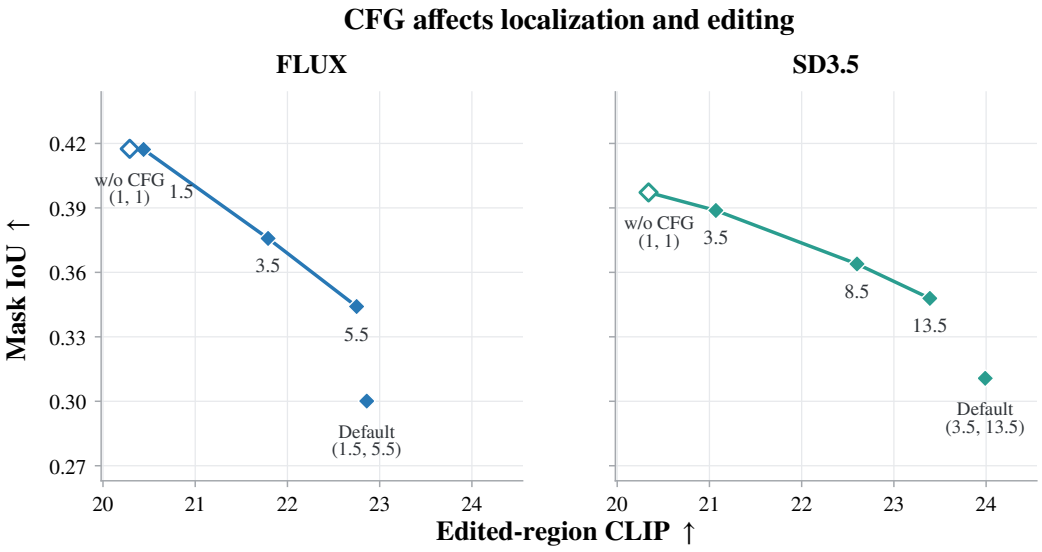

Figure 10: Additional CFG comparisons on FLUX and SD3.5. Point labels give symmetric guidance strength g; the w/o CFG and default CFG points show their complete (source, target) settings. Connected points form the symmetric guidance sweep. All edited images are generated without a spatial mask; mask IoU evaluates the prior extracted from the corresponding velocity differences.

E.3 Effect of Prior-Extraction CFG

We evaluate this ablation on the same 556 local-edit tasks in PIE-Bench, excluding tasks whose ground-truth masks cover the entire image. All seven metrics, including both CLIP scores, are computed on this common subset. We compare a prior-extraction stage using w/o CFG $(1, 1)$ with one using the backbone's default CFG: $(1.5, 5.5)$ for FLUX and $(3.5, 13.5)$ for SD3/SD3.5. The prior-guided editing stage uses default CFG in both cases, with the same downstream sampling settings within each backbone. Both prior-extraction configurations use the full schedules and aggregation

windows specified in Appendix E.1, with $\tau = 0.55$ and $s = 0.08$. The resulting soft prior is fixed throughout the editing stage. FLUX uses saved 8-bit soft masks, while SD3/SD3.5 use float32 priors.

Table 8: Effect of prior-extraction CFG with prior-guided editing fixed at default CFG. All metrics are evaluated on the same 556 local-edit tasks in PIE-Bench. Metric scaling follows Appendix E.1.

Backbone	Prior CFG	Structure ↓	Background preservation				CLIP ↑	
			PSNR ↑	LPIPS ↓	MSE ↓	SSIM ↑	Whole	Edited
FLUX	w/o CFG	7.860	30.538	20.895	19.225	94.544	25.133	21.231
	default CFG	12.179	25.968	41.770	45.080	91.744	25.256	21.228
SD3	w/o CFG	10.441	27.991	31.817	28.740	91.695	26.186	22.095
	default CFG	15.647	24.828	54.023	54.934	89.167	26.367	22.121
SD3.5	w/o CFG	9.163	28.783	29.214	24.850	92.030	26.319	22.106
	default CFG	13.389	25.673	48.216	45.470	89.836	26.586	22.129

As shown in Table 8, extracting priors under w/o CFG reduces background LPIPS by 50.0%, 41.1%, and 39.4% on FLUX, SD3, and SD3.5, respectively, and improves all other preservation metrics. The corresponding relative changes in edited-region CLIP are $+0.013\%$, -0.121%, and -0.103%. All relative changes use the corresponding default-CFG-prior result as the reference. These results support using w/o CFG for prior extraction while retaining default CFG for semantic editing.

E.4 Per-step Global Norm Matching

We test whether a single global rescaling of the w/o CFG update can recover default CFG editing. At every step of the norm-matched trajectory, we evaluate the w/o CFG update $\mathbf{U}_t$ and default CFG update $\mathbf{H}_t$ at the *same current source/target states and with the same sampled noise*. We advance the state using only

$$\widehat{\mathbf{U}}_t = \frac{\|\mathbf{H}_t\|_{\mathrm{F}}}{\|\mathbf{U}_t\|_{\mathrm{F}} + 10^{-8}} \mathbf{U}_t. \tag{13}$$

The Frobenius norm covers all spatial positions and channels, yielding one scalar for the entire update. Both updates are recomputed at the next state; the reference magnitude is not taken from a separately generated default-CFG trajectory. We use the same sampling settings as in Table 7.

Table 9: Direct editing with w/o CFG, global norm matching, and default CFG. Both CLIP metrics use the actual outputs of all 700 tasks; background and structure metrics use the 556 local tasks. The scale factors are those specified in Appendix E.1.

Backbone	Update	Structure ↓	Background preservation				CLIP ↑	
			PSNR ↑	LPIPS ↓	MSE ↓	SSIM ↑	Whole	Edited
FLUX	w/o CFG	0.95	36.30	7.77	4.70	96.32	23.47	20.29
	Norm-matched	42.84	20.34	95.92	142.43	85.35	25.13	21.30
	default CFG	28.99	21.72	104.79	98.32	83.87	25.88	22.71
SD3	w/o CFG	0.97	35.32	10.23	6.11	94.31	23.49	20.33
	Norm-matched	84.07	16.76	178.42	298.79	75.39	24.50	20.67
	default CFG	26.39	22.27	100.15	85.71	83.93	27.51	23.92
SD3.5	w/o CFG	0.96	35.26	10.48	6.09	94.22	23.49	20.34
	Norm-matched	64.17	18.39	149.86	214.27	78.70	24.60	20.80
	default CFG	22.55	23.27	88.78	69.66	85.57	27.62	23.95

Global norm matching increases edited-region CLIP relative to w/o CFG but remains below default CFG on all three backbones, while structure distance is higher (Table 9). SD3 and SD3.5 also deteriorate in all four background metrics; FLUX has mixed background results. The norm-matched

Table 10: Multi-seed quantitative results on PIE-Bench. We report mean ± standard deviation over five runs with different random seeds. Lower is better for Structure Distance, LPIPS, and MSE; higher is better for PSNR, SSIM, and CLIP similarity.

Method	Backbone	Structure	Background Preservation				CLIP Similarity	
		Distance ↓	PSNR ↑	LPIPS$_{\times 10^3}$ ↓	MSE$_{\times 10^4}$ ↓	SSIM$_{\times 100}$ ↑	Whole ↑	Edited ↑
FlowEdit	FLUX	28.99 ± 0.24	21.72 ± 0.06	104.79 ± 0.06	98.32 ± 0.26	83.87 ± 0.02	25.88 ± 0.01	22.71 ± 0.04
Ours	FLUX	7.86 ± 0.21	30.54 ± 0.05	20.90 ± 0.05	19.23 ± 0.22	94.54 ± 0.02	25.25 ± 0.01	22.15 ± 0.03
FlowEdit	SD3	26.39 ± 0.26	22.27 ± 0.07	100.15 ± 0.07	85.71 ± 0.29	83.93 ± 0.03	27.51 ± 0.02	23.92 ± 0.04
Ours	SD3	10.01 ± 0.22	28.16 ± 0.05	31.34 ± 0.05	27.85 ± 0.22	91.75 ± 0.02	26.71 ± 0.01	23.37 ± 0.03
FlowEdit	SD3.5	22.55 ± 0.25	23.27 ± 0.07	88.78 ± 0.06	69.66 ± 0.27	85.57 ± 0.02	27.62 ± 0.02	23.95 ± 0.04
Ours	SD3.5	8.90 ± 0.20	28.78 ± 0.05	28.79 ± 0.05	24.59 ± 0.21	92.03 ± 0.02	26.86 ± 0.01	23.35 ± 0.03

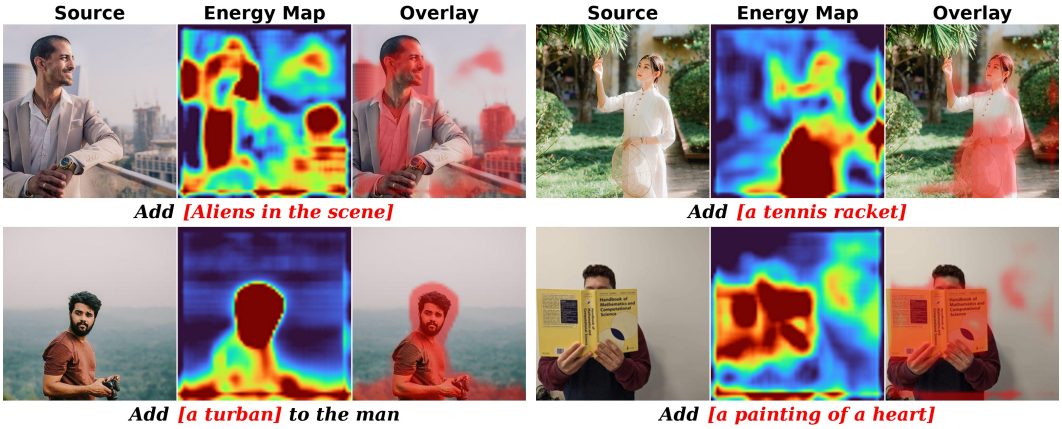

Figure 11: Limitations of spatial-prior extraction in object-add edits using FLUX.

edits exhibit the severe blur shown in Figure 4. Matching total update magnitude is therefore insufficient to recover the editing quality of default CFG.

F Multi-seed Main Results

To assess the stability of the quantitative results, we repeat the main experiments five times with different random seeds. The random seeds affect the stochastic components of the editing process, including the noise samples used in FlowEdit-style updates and prior extraction. We report the mean and the run-to-run standard deviation over these five runs. The repeated experiments follow the same evaluation protocol as the main paper, using PIE-Bench and the same metrics for structure distance, background preservation, and semantic editing fidelity.

Across seeds, our method consistently improves background and structure preservation over the corresponding FlowEdit baseline. The standard deviations are small relative to the performance gains, indicating that the improvement is not caused by a particular random seed. The semantic editing metrics remain competitive, suggesting that the spatial reweighting improves preservation without substantially weakening the intended edit.

G Limitations

Our method is most reliable for local edits where the source image already contains a clear editing target or spatial anchor. In such cases, temporally aggregating FlowEdit velocity differences $\Delta \mathbf{V}_t$ can accumulate edit-driven responses around the relevant region and produce a selective spatial prior. This reliance on a spatial anchor also reveals several limitations.

First, for *object-add* edits, the target object usually has no clear source-side counterpart or spatial anchor in the source image. As a result, the edit-driven updates lack a stable region on which to accumulate across timesteps. The extracted prior may therefore become diffuse or attach to

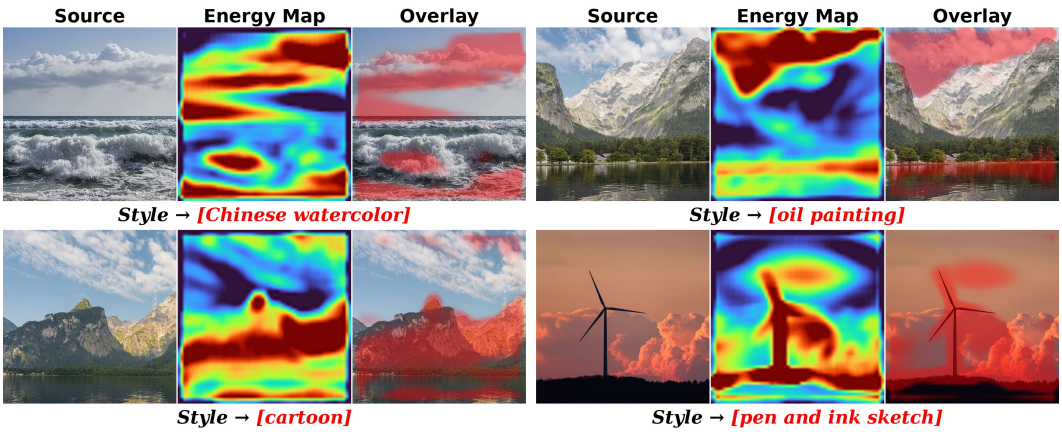

Figure 12: Limitations of spatial-prior extraction in style-change edits using FLUX.

semantically related existing regions, rather than accurately localizing where the newly added object should appear, as illustrated in Figure 11.

Second, for *style-change* edits, the target modification is often a global or large-range appearance transformation, such as changing texture, color, brushstroke, or overall rendering style. Such edits do not necessarily require a localized mask, because the desired change is expected to affect the whole image relatively consistently. Accordingly, our temporal aggregation mechanism is less suitable for this setting: the resulting prior does not uniformly cover the image, but is instead influenced by semantic structures, local textures, and high-response regions, leading to a content-dependent and non-uniform distribution, as shown in Figure 12. Therefore, these cases should not be interpreted simply as localization failures: our prior is deliberately designed for local, object-related, or spatially anchored edits, and style changes that require globally consistent modification are outside the scope of this method. For such edits, we recommend applying raw FlowEdit, whose spatially global velocity differences naturally support global transformations.

Our method also introduces one additional prior-extraction pass. Although this pass does not require training, optimization, external segmentation, or attention manipulation, it still increases inference time compared with the original FlowEdit (approximately $2\times$ the sampling steps, or $1.25\times$ with the short schedule; Appendix C).

H Broader Impact

This work improves training-free text-guided image editing by reducing unintended changes in background and non-edit regions. This can benefit creative editing, visual content production, and user-controlled image manipulation, especially where preserving source details is important.

However, more reliable image editing tools may also be misused to create misleading, manipulated, or deceptive visual content. The proposed method does not introduce a new generative model, but it can improve the controllability of existing pretrained image generation backbones. We encourage responsible use of the method and adherence to the usage policies and safety guidelines of the underlying pretrained generative models.

I Assets and Licenses

We use publicly available datasets, pretrained models, and baseline methods. The main benchmark is PIE-Bench, which provides source images, source and target prompts, editing instructions, and ground-truth edit masks for text-guided image editing evaluation. We use publicly available pretrained flow backbones, including FLUX, SD3, and SD3.5, following their respective licenses and terms of use. We also compare with publicly described training-free editing baselines, including FlowEdit and recent flow-based editing methods.

All existing assets used in this paper are cited in the main text or references. The proposed method does not introduce a new dataset or a new pretrained generative model. The released asset is the anonymized implementation of our method, with instructions for reproducing the main experiments.