GRAPE: RETHINKING THE VELOCITY FIELD AND INTEGRATION FOR INVERSION-FREE FLOW EDITING

Anonymous authors
Paper under double-blind review

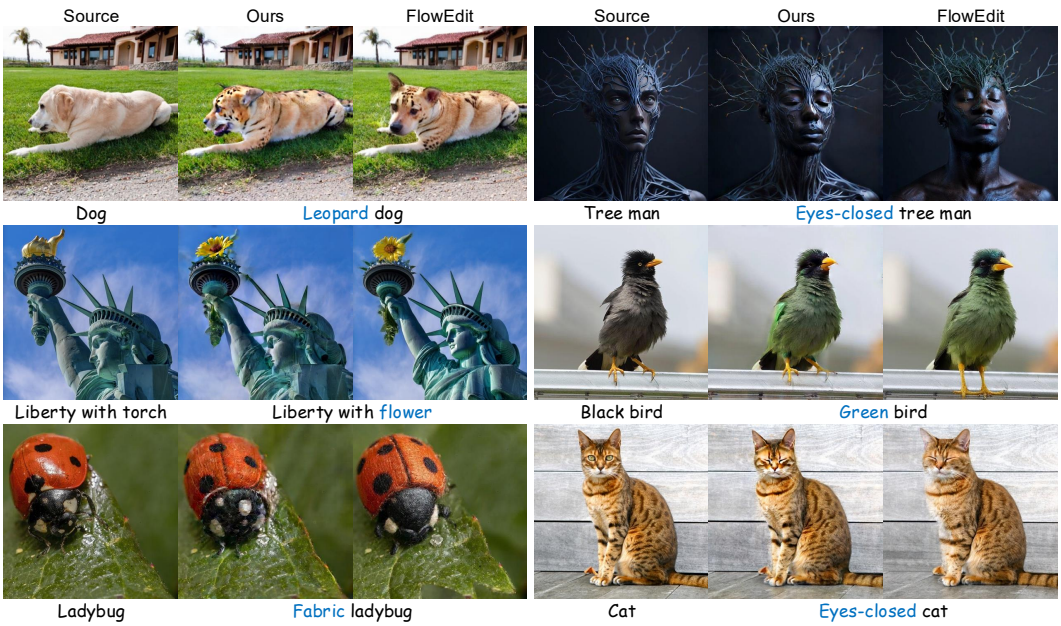


Figure 1: **GRAPE enables targeted edits while preserving unrelated image content.** Examples show the source image, FlowEdit (Kulikov et al., 2025) with 132 NFEs, and GRAPE with 30 NFEs. Across diverse edits, GRAPE realizes the requested changes with fewer unintended alterations.

ABSTRACT

Recent inversion-free image editing methods based on pretrained flow models directly modify source images using differences between source- and target-conditional velocities. However, this process often introduces changes beyond the requested attributes. We revisit these editing dynamics in terms of how the editing velocity is constructed and integrated across noise levels. Since the source- and target-conditional velocities are evaluated at distinct noisy states, their difference mixes the response to the condition change with a denoising response to the state difference, which can affect content that should be preserved. Based on this observation, we construct a gradient-aligned field that evaluates both conditions at a shared noisy observation of the current edit. Under population-optimal velocities, this field is proportional to the target-to-source log-density-ratio gradient. We further find that standard noise-schedule integration concentrates editing progress in late, low-noise steps, revealing a mismatch between noise-schedule progression and clean-state progress. This motivates progress-adaptive integration, which measures and compensates for the mismatch between noise-schedule progression and clean-state progress. We introduce GRAPE (Gradient-Aligned, Progress-adaptive Editing), a training-free and inversion-free method that combines gradient-aligned field construction with progress-adaptive integration. Across three flow backbones on PIE-Bench, GRAPE consistently improves the trade-off between target alignment and source preservation and retains this advantage under substantially reduced inference budgets.

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

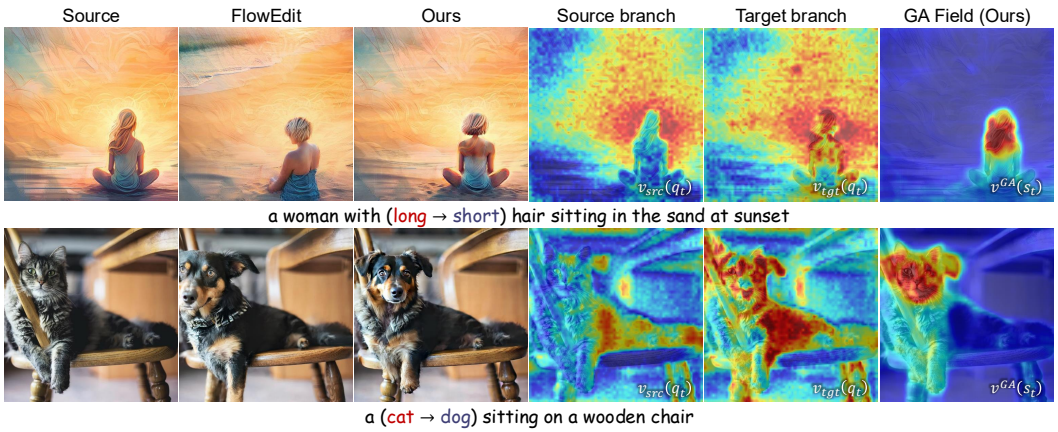


Figure 2: **Unintended changes and accumulated velocity responses.** FlowEdit evaluates source and target velocities at different noisy observations, producing responses in regions to be preserved and unintended changes in orientation and background. The gradient-aligned (GA) field evaluates both conditions at a shared observation, yielding more localized responses and better preservation. Heatmaps accumulate response magnitudes over all steps. Adaptive integration is disabled.

1 Introduction

Text-guided image editing aims to modify the attributes specified by a user while preserving the remaining content of the source image (Meng et al., 2022; Hertz et al., 2023; Tumanyan et al., 2023). Pretrained diffusion and flow models enable training-free editing through their learned visual and semantic priors (Ho et al., 2020; Song et al., 2021b; Rombach et al., 2022; Lipman et al., 2023; Liu et al., 2023; Esser et al., 2024). Conventional inversion-based methods first invert the source image into its corresponding noise and then regenerate it under the target condition (Song et al., 2021a; Mokady et al., 2023; Cao et al., 2023; Rout et al., 2025; Wang et al., 2025; Avrahami et al., 2025; Xie et al., 2025). This inversion and regeneration process incurs additional computational cost and may introduce reconstruction errors (Wallace et al., 2023; Ju et al., 2024). These limitations have motivated inversion-free methods that directly update the input image without mapping it to noise (Xu et al., 2024; Kulikov et al., 2025).

FlowEdit (Kulikov et al., 2025), a representative inversion-free approach, keeps the clean editing state and updates it using the difference between source- and target-conditioned predictions. The two branches are evaluated at different noisy observations constructed at progressively lower noise levels. Subsequent work has addressed complementary aspects of inversion-free editing, including source consistency, edit-strength control, and velocity correction (Kim et al., 2026; Endo et al., 2026; Beaudouin et al., 2026). Despite these advances, these methods can still alter content beyond the requested edit, compromising preservation of the remaining source content. In Figure 2, a request to shorten the person's hair also changes their seated orientation and the surrounding background, although the instruction specifies neither change.

To understand these unintended changes, we examine the evaluation states of the two conditional velocities used to construct the editing direction. A flow model's conditional velocity specifies the local direction of transport from a noisy state along the condition-specific probability path. The predicted velocity therefore varies with the evaluation state, even under a fixed condition (Lipman et al., 2023; Liu et al., 2023). In existing methods, the source and target branches differ in both the condition and the evaluation state. As a result, their velocity difference reflects not only the change in condition but also a denoising response to the difference in evaluation state. This state-induced component can introduce responses in regions that should remain unchanged, such as pose and background, potentially contributing to unintended edits. The accumulated response maps in Figure 2 show discrepant responses between the two branches in these regions, consistent with this interpretation.

Motivated by this observation, we define the editing direction as the difference between the velocities predicted under the source and target conditions at a shared noisy observation of the current edit.

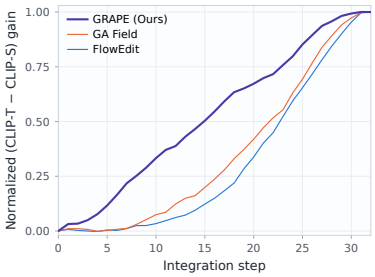


Figure 3: **Accumulation of target-relative editing progress.** Progress is measured by the CLIP margin between the target and source prompts, normalized by the final value of each curve. Under standard integration, both FlowEdit and the GA field concentrate progress in later, low-noise steps, whereas GRAPE, which applies progress-adaptive integration to the GA field, distributes it more evenly without additional model evaluations.

This comparison holds the evaluation state fixed, isolating the effect of the condition change. We show that, under population-optimal velocities, this direction is proportional to the target-to-source log-density-ratio gradient, locally increasing target-relative preference (Dhariwal & Nichol, 2021; Ho & Salimans, 2022). Rather than deriving the edit field from a relation between the source and target trajectories, we defined it directly from the current edit. Empirically, the resulting changes are more localized to the intended region and cause fewer unintended changes.

We next examine how to integrate the editing direction over the noise schedule. Since clean-space updates use velocities evaluated at noisy observations, editing progress can lag behind the noise schedule. We find that standard integration concentrates much of this progress in late, low-noise stages (Figure 3), leaving the edit behind the schedule for much of the process. Consequently, the edit may reach late noise levels before sufficient target-relative change has been realized. To reduce this gap, we propose progress-adaptive integration, which increases the update along the same editing direction only when the edit falls behind a progress reference aligned with the noise schedule. Together, these components form GRAPE a training-free and inversion-free framework that revisits both how the editing direction is constructed and how it is applied.

Across SD3 (Esser et al., 2024), FLUX.1-dev (Black Forest Labs, 2024), and Qwen-Image (Wu et al., 2025) on PIE-Bench (Ju et al., 2024), GRAPE consistently improves the trade-off between target alignment and source preservation. Our component analyses support the improved localization of the gradient-aligned field and the complementary effect of progress-adaptive integration, while reduced-step experiments show that these gains persist under substantially lower computation.

2 Preliminaries: Inversion-Free Flow Editing

We consider a pretrained text-conditioned rectified flow model (Liu et al., 2023; Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023) with velocity $v_\theta(x, \sigma; c)$, where c is a text condition and $\sigma \in [0, 1]$ denotes flow time from clean data ($\sigma = 0$) to Gaussian noise ($\sigma = 1$). The model follows the linear probability path $x_\sigma = (1 - \sigma)x_0 + \sigma\epsilon$ with $x_0 \sim p_0(\cdot \mid c)$ and $\epsilon \sim \mathcal{N}(0, I)$, and is trained to predict its velocity $\epsilon - x_0$. The corresponding clean prediction is $\widehat{x}_0(x, \sigma; c) = x - \sigma v_\theta(x, \sigma; c)$. We abbreviate the source- and target-conditional velocities as $v_{\mathrm{src}}(x, \sigma) = v_\theta(x, \sigma; c_{\mathrm{src}})$ and $v_{\mathrm{tgt}}(x, \sigma) = v_\theta(x, \sigma; c_{\mathrm{tgt}})$, which in practice include classifier-free guidance (Ho & Salimans, 2022).

FlowEdit (Kulikov et al., 2025) derives a direct editing trajectory from an inversion-based formulation. Let x_σ^{src} denote the source trajectory obtained by inverting x_{src} along the source-conditional flow, and let x_σ^{tgt} denote the target trajectory generated from the same Gaussian endpoint along the target-conditional flow. The corresponding editing trajectory is

$$z_\sigma^{\mathrm{inv}} = x_{\mathrm{src}} + x_\sigma^{\mathrm{tgt}} - x_\sigma^{\mathrm{src}}. \tag{1}$$

Since the two trajectories share their noise endpoint, $z_1^{\mathrm{inv}} = x_{\mathrm{src}}$, while $z_0^{\mathrm{inv}} = x_0^{\mathrm{tgt}}$ is the edited image. The flow time σ thus indexes both the noise level of the two generative trajectories and the progress of the edit from the source to the target.

FlowEdit avoids inversion by replacing the source trajectory with a stochastic source-side state. At step i, it samples $\epsilon_i \sim \mathcal{N}(0, I)$ and constructs $q_i = (1 - \sigma_i)x_{\mathrm{src}} + \sigma_i\epsilon_i$. The displacement accumulated in the clean editing state z_i is then transferred to the target side as $p_i = q_i + z_i - x_{\mathrm{src}}$. Writing q_σ and p_σ for these states in continuous time, the sampled reverse-time field and the resulting

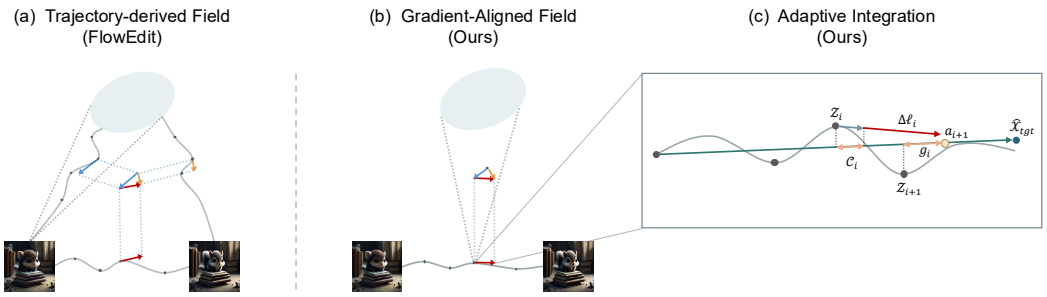


Figure 4: **Overview of GRAPE.** At each step, GRAPE forms a shared noisy observation of the current edit and evaluates both conditions to obtain the gradient-aligned (GA) field. Progress-adaptive integration compares the nominal update with a noise-schedule-aligned reference and lengthens the step only when the edit falls behind, without changing the GA direction.

dynamics are

$$u_\sigma^{\mathrm{FE}} = v_{\mathrm{src}}(q_\sigma, \sigma) - v_{\mathrm{tgt}}(p_\sigma, \sigma), \qquad \frac{dz_\sigma}{d\sigma} = -\mathbb{E}_\epsilon\left[u_\sigma^{\mathrm{FE}}\right] = \mathbb{E}_\epsilon\left[v_{\mathrm{tgt}}(p_\sigma, \sigma) - v_{\mathrm{src}}(q_\sigma, \sigma)\right]. \quad (2)$$

The dynamics are initialized at $z_{\sigma_{\max}} = x_{\mathrm{src}}$, where $\sigma_{\max} < 1$ sets the noise level at which editing begins, and integrated toward $\sigma = 0$, with the expectation approximated using sampled noise. The source and target velocities are evaluated at two different noisy states, which differ by the accumulated edit $z_\sigma - x_{\mathrm{src}}$. We refer to editing fields constructed in this way as trajectory-derived, since they are derived from the trajectory relation in equation 1.

3 Rethinking Inversion-Free Dynamics

Our analysis focuses on two aspects of these dynamics, velocity-field construction and integration. The former leads to a gradient-aligned field derived from target-relative preference (Section 3.1), while the latter motivates progress-adaptive integration based on editing progress in the clean state (Section 3.2). Figure 4 provides an overview of GRAPE.

3.1 Velocity Field

Since inversion-free editing directly updates a clean editing state, we define the editing direction by how it increases the target-relative preference of the current state. Rather than deriving the field from a relation between source and target trajectories, we compare both conditions at a shared noisy observation of the current edit.

3.1.1 Gradient-Aligned Field

In trajectory-derived fields, the source and target velocities are evaluated at q_i and p_i, which differ by the accumulated edit $z_i - x_{\mathrm{src}}$ (Section 2). Since a conditional velocity depends on its evaluation state even under a fixed condition, and both the condition and the state change from q_i to p_i, their difference cannot separate the response to the condition change from that to the state difference. We make this decomposition explicit in Section 3.1.2.

Isolating the effect of the condition change therefore requires evaluating both conditions at the same state. A natural choice for this state is a noisy observation of the current edit, since it is the quantity being updated and reflects the edit accumulated so far. We retain the noise schedule and the linear probability path of the preceding dynamics, and at noise level σ_i define the reverse-time editing direction as

$$s_i = (1 - \sigma_i) z_i + \sigma_i \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, I), \qquad u_i^{\mathrm{GA}} = v_{\mathrm{src}}(s_i, \sigma_i) - v_{\mathrm{tgt}}(s_i, \sigma_i). \quad (3)$$

Since s_i is constructed from the current z_i at every step, both conditional velocities are evaluated at the same observation of the evolving edit. We refer to u_i^{GA} as the gradient-aligned (GA) field, whose gradient interpretation is established in Proposition 1.

Target-relative preference. We define the target-relative preference of the current edit as the target-to-source log-density ratio at the shared observation s_i,

$$F_i(z_i; \epsilon_i) = \log \frac{p_{\sigma_i}(s_i \mid c_{\mathrm{tgt}})}{p_{\sigma_i}(s_i \mid c_{\mathrm{src}})}, \tag{4}$$

where $p_{\sigma_i}(\cdot \mid c)$ denotes the conditional marginal density at noise level σ_i. Since s_i is a noisy observation of z_i, F_i quantifies the target-relative preference of the current edit at noise level σ_i, and approaches the preference of z_i itself as σ_i decreases. For fixed ϵ_i and σ_i, $\nabla_{z_i} F_i$ gives the direction of steepest local increase in this preference.

Proposition 1. *For each condition c, let $v_c^*(x, \sigma) = \mathbb{E}[\epsilon - x_0 \mid x_\sigma = x, c]$ denote the population-optimal velocity of the Gaussian linear probability path, and write $v_{\mathrm{src}}^* = v_{c_{\mathrm{src}}}^*$ and $v_{\mathrm{tgt}}^* = v_{c_{\mathrm{tgt}}}^*$. Then, for $0 < \sigma_i < 1$ and fixed ϵ_i,*

$$\nabla_{z_i} F_i = \frac{(1-\sigma_i)^2}{\sigma_i} \left[v_{\mathrm{src}}^*(s_i, \sigma_i) - v_{\mathrm{tgt}}^*(s_i, \sigma_i) \right]. \tag{5}$$

Evaluating both velocities at the same observation cancels their condition-independent terms, leaving the difference of the conditional scores (Song et al., 2021b). The proof is given in Appendix B. Since the coefficient in equation 5 is positive, the population-optimal GA field is aligned with $\nabla_{z_i} F_i$. As the pretrained velocities approximate their population-optimal counterparts, u_i^{GA} approximates this direction in practice.

Continuous-time dynamics. The corresponding mean-field continuous-time dynamics are

$$\frac{dz_\sigma}{d\sigma} = -\mathbb{E}_\epsilon \left[u_\sigma^{\mathrm{GA}} \right] = \mathbb{E}_\epsilon \left[v_{\mathrm{tgt}}(s_\sigma^\epsilon, \sigma) - v_{\mathrm{src}}(s_\sigma^\epsilon, \sigma) \right], \qquad s_\sigma^\epsilon = (1-\sigma) z_\sigma + \sigma \epsilon. \tag{6}$$

These dynamics share the form of equation 2, including its initialization and integration, and differ only in the evaluation states of the two velocities. Here, z_σ denotes the clean editing state indexed by the noise schedule rather than a noisy state along the generative flow trajectory, a distinction we return to in Section 3.2. Evaluating both conditions at the same observation also lets their classifier-free guidance reuse a single unconditional prediction, saving one model evaluation per step on backbones with explicit guidance.

3.1.2 Comparison with Trajectory-Derived Fields

The density-ratio perspective also applies to the trajectory-derived field of Section 2. Let $u_i^{\mathrm{FE},*}$ denote the FlowEdit field under the population-optimal velocities. Adding and subtracting the source-conditional velocity at the target-side state p_i gives

$$u_i^{\mathrm{FE},*} = \underbrace{\frac{\sigma_i}{1-\sigma_i} \nabla_{p_i} \log \frac{p_{\sigma_i}(p_i \mid c_{\mathrm{tgt}})}{p_{\sigma_i}(p_i \mid c_{\mathrm{src}})}}_{\text{target-relative steering at } p_i} + \underbrace{v_{\mathrm{src}}^*(q_i, \sigma_i) - v_{\mathrm{src}}^*(p_i, \sigma_i)}_{\text{source-field correction}}. \tag{7}$$

This decomposition mirrors the steering and fidelity terms in FlowSlider (Endo et al., 2026). The first term provides target-relative steering at p_i through the target-to-source log-density-ratio gradient. The second term compares the same source condition at q_i and p_i, capturing a response induced by the difference evaluation state. The trajectory-derived field can therefore be decomposed into target-relative steering and a cross-state source-field correction.

The GA field consists of the steering term alone, evaluated at s_i, a noisy observation of the current edit itself. The trajectory-derived steering term is instead evaluated at $p_i = s_i + \sigma_i(z_i - x_{\mathrm{src}})$, which departs from this observation in proportion to the accumulated edit, and is accompanied by a source-field correction not implied by the target-relative objective. We examine the effect of this difference on spatial localization in Section 4.4.

3.2 INTEGRATION

We next examine how the GA field should be integrated over the editing process. In particular, we ask whether target-relative progress in the clean editing state remains synchronized with the progression of the noise schedule. This analysis motivates our progress-adaptive integration.

3.2.1 NOISE-SCHEDULE PROGRESS AND CLEAN-STATE RESPONSE

In inversion-based editing, the noisy latent evolves directly along the noise schedule, coupling the evaluation state to flow time along a single trajectory. Inversion-free dynamics instead evaluate the velocity at a noisy observation and apply it to update a separate clean editing state. Consequently, progress through the noise schedule need not translate into comparable target-relative progress in the clean state. Insufficient early updates can leave the clean state behind the stage represented by the noise schedule.

We use the CLIP margin (Radford et al., 2021), defined as the difference between similarities to the target and source prompts, as a measure of target-relative progress. As shown in Figure 3, both FlowEdit and the GA field accumulate little margin during early and intermediate steps, with most progress occurring in the late, low-noise stages. This indicates a mismatch between noise-schedule progression and the semantic progress realized in the clean editing state. Since the schedule advances independently of this accumulated progress, the clean state may reach later noise levels before sufficient target-relative change has been realized.

3.2.2 CLEAN-STATE REFERENCE AND ADAPTIVE INTEGRATION

To measure the lag between the accumulated edit and the noise schedule, we define a clean-state progress reference. Under the shared-noise linear paths in Section 2, the ideal inversion-based editing trajectory interpolates between the source and target as

$$a^{\star}_{i+1} = \sigma_{i+1} x_{\mathrm{src}} + (1 - \sigma_{i+1}) x_{\mathrm{tgt}}.$$

Since x_{tgt} is unavailable during editing, we estimate its displacement using the target-conditioned clean prediction at the shared observation,

$$\widehat{x}_{\mathrm{tgt},i} = s_i - \sigma_i v_{\mathrm{tgt}}(s_i, \sigma_i), \quad d_i = \widehat{x}_{\mathrm{tgt},i} - x_{\mathrm{src}}, \quad a_{i+1} = x_{\mathrm{src}} + (1 - \sigma_{i+1}) d_i. \tag{8}$$

Here, d_i estimates the source-to-target displacement, while a_{i+1} places this displacement at the stage specified by σ_{i+1}.

We first form the standard update $z^{\mathrm{raw}}_{i+1} = z_i + \ell_i u^{\mathrm{GA}}_i$, where $\ell_i = \sigma_i - \sigma_{i+1} > 0$. Since a_{i+1} is constructed from an estimated target endpoint, we compare it with the standard update only along the predicted displacement d_i. We define the directional deficit and the directional progress per unit application length as

$$g_i = \left\langle a_{i+1} - z^{\mathrm{raw}}_{i+1}, d_i \right\rangle, \qquad c_i = \left\langle u^{\mathrm{GA}}_i, d_i \right\rangle.$$

A positive g_i indicates that the standard update falls behind the reference along d_i, while a positive c_i indicates that extending the GA field can reduce this deficit. We therefore define

$$\Delta\ell_i = \begin{cases} g_i / c_i, & g_i > 0 \text{ and } c_i > 0, \\ 0, & \text{otherwise}, \end{cases} \qquad z_{i+1} = z_i + (\ell_i + \Delta\ell_i)\, u_i. \tag{9}$$

The ratio g_i/c_i gives the additional application length required to close the directional deficit while preserving the GA direction and the evaluation noise schedule. In practice, we add a small constant ε to the denominator and cap the total application length at $M\ell_i$ for stability. The reference reuses the target velocity already computed for the GA field and therefore requires no additional velocity evaluations. The complete procedure is summarized in Algorithm 1 in Appendix C.

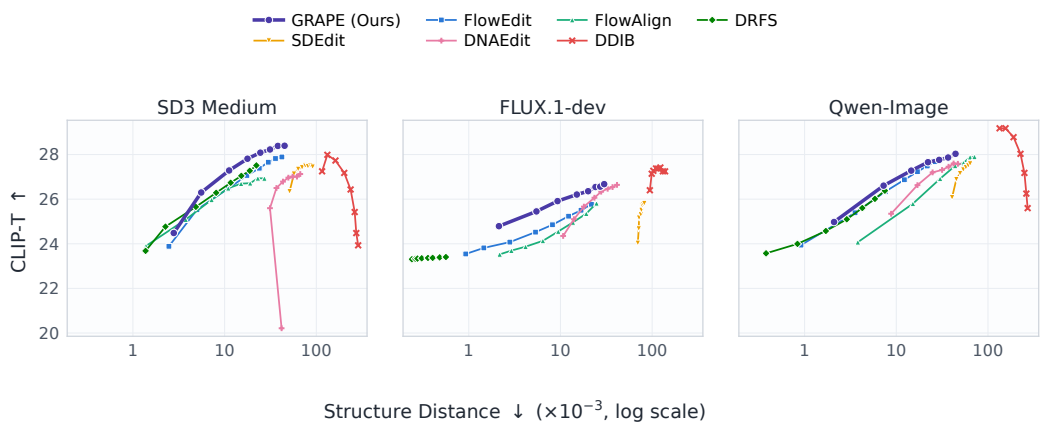


Figure 5: **Editability and preservation trade-offs across flow backbones.** Each curve traces a target-guidance sweep on PIE-Bench for SD3 Medium, FLUX.1-dev, and Qwen-Image from left to right. Higher CLIP-T and lower Structure Distance are better.

4 Experimental Results

4.1 Experimental Setup

We evaluate on the 700 images of PIE-Bench (Ju et al., 2024) with SD3 Medium (Esser et al., 2024), FLUX.1-dev (Black Forest Labs, 2024), and Qwen-Image (Wu et al., 2025), against SDEdit (Meng et al., 2022), DDIB (Su et al., 2023), DNAEdit (Xie et al., 2025), the inversion-free FlowEdit (Kulikov et al., 2025) and FlowAlign (Kim et al., 2026), and the distillation-based DRFS (Beaudouin et al., 2026), all from their official implementations. Following the protocol of prior inversion-free work (Kulikov et al., 2025; Kim et al., 2026), every method shares the same sampling configuration within each backbone, taking 33 steps of a T=50 schedule on SD3 and Qwen-Image and 24 of T=28 on FLUX.1-dev, with source guidance 3.5 on SD3 and 1.5 otherwise. We set M to 3 except on the guidance-distilled FLUX.1-dev, where it is 6. Target guidance is swept over the range in which each backbone remains stable, and alignment is measured by CLIP-T (Radford et al., 2021) and preservation by Structure Distance (Tumanyan et al., 2022), PSNR, LPIPS (Zhang et al., 2018), and SSIM (Wang et al., 2004) over the unedited regions. Since weaker edits trivially improve preservation, single-operating-point results compare methods at matched edit strength, using for each method the configuration closest to CLIP-T 26.5, a level within the range covered by all compared methods.

4.2 Quantitative Results

Editability and preservation trade-off. Figure 5 compares CLIP-T and Structure Distance across the full target-guidance sweep. GRAPE forms the upper-left envelope on all three backbones, preserving the source structure better at comparable CLIP-T and achieving stronger alignment at comparable Structure Distance. Its consistent gain over FlowEdit across SD3, FLUX.1-dev and Qwen-Image indicate that the improvement is not specific to a particular architecture. Trade-off curves for LPIPS, PSNR, and SSIM are provided in the Appendix E.

Perceptual and human-aligned evaluation. We further evaluate DreamSim (Fu et al., 2023), a perceptual distance calibrated to human similarity judgments, and EditHF (Xu et al., 2026), which scores alignment, preservation, and visual quality with a model trained on human editing preferences. On SD3 at matched edit strength (Table 1), GRAPE achieves the lowest DreamSim distance and the highest EditHF preservation and visual-quality scores, with alignment close to the best baseline. Results on FLUX.1-dev and Qwen-Image are reported in Appendix E.

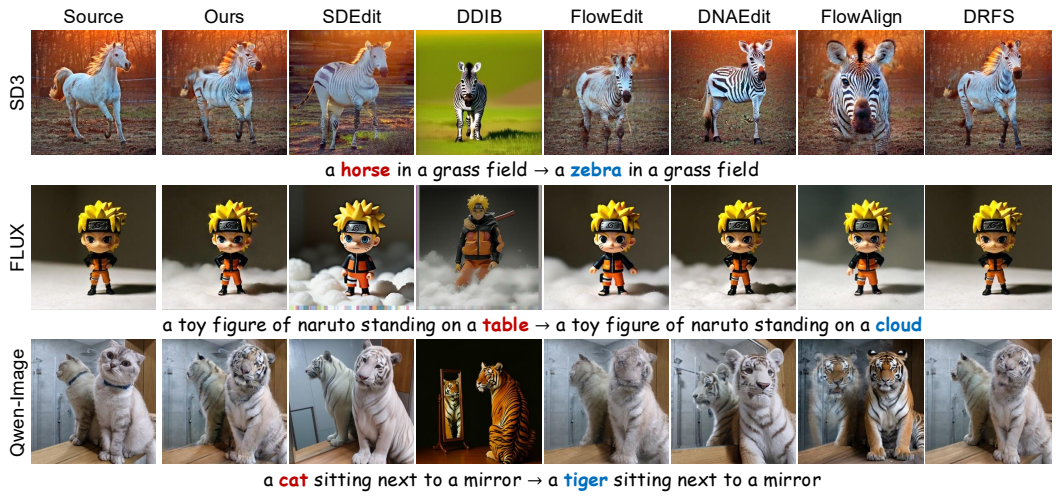


Figure 6: **Qualitative comparison on SD3, FLUX.1-dev, and Qwen-Image.** Methods are compared at matched CLIP-T. In each prompt, removed words are shown in red and added words in blue. Competing methods often alter the orientation, composition, or identity of the source content, whereas GRAPE better preserves unedited content. Remaining baselines are shown in Appendix F.

Table 1: **Perceptual and human-aligned evaluation on SD3 at matched edit strength.** EHF-A, EHF-P, and EHF-V denote EditHF alignment, preservation, and quality scores (Xu et al., 2026).

Method	CLIP-T	Structure ↓	DreamSim ↓	EHF-A ↑	EHF-P ↑	EHF-V ↑
SDEdit	26.35	51.12	0.2882	50.53	39.71	40.64
DDIB	26.43	236.58	0.5723	46.18	34.78	32.35
FlowEdit	26.48	10.99	0.1358	**53.44**	51.75	50.55
DNAEdit	26.50	36.74	0.2769	51.17	41.62	41.77
FlowAlign	26.49	10.97	0.1619	53.29	51.54	49.87
DRFS	26.29	8.09	0.1082	52.87	53.10	51.18
GRAPE (Ours)	26.56	**6.54**	**0.1034**	52.95	**54.00**	**51.44**

4.3 Qualitative Results

Figure 6 compares editing results across representative categories and backbones at matched edit strength. While all methods apply the requested change, competing methods often alter content beyond it, whereas GRAPE better preserves source-specific content such as orientation, geometry, and background structure. These examples visually support the trade-offs observed in Figure 5.

4.4 Ablation and Analysis

Component contributions. We isolate the two components on SD3 by first replacing the trajectory-derived field of FlowEdit with the GA field and then adding progress-adaptive integration to obtain GRAPE. Figure 7 shows that the GA field substantially reduces structural changes within the shared guidance range, occupying the low-distortion region of the trade-off. Adaptive integration then strengthens the edit at the same guidance scales, reaching higher target alignment while largely retaining this preservation, so that the curve moves outward over most of the range. The two components are therefore complementary, one constructing a direction that reflects the condition change and the other applying it to the extent the schedule assumes.

Editing-field localization. We next examine whether the GA field concentrates the editing response within the intended region. We compare FlowEdit and the GA field at matched edit strength, using the same noise schedule and noise samples with adaptive integration disabled. We accumulate

Table 2: **Performance under reduced inference budgets on SD3.** Each cell reports FlowEdit / GRAPE. T denotes the schedule discretization, N_{edit} the number of editing updates, and NFE the number of velocity network evaluations. GRAPE outperforms FlowEdit at every budget.

T/N_{edit}	NFE	CLIP-T ↑	Structure ↓	PSNR ↑	LPIPS ↓
50/33	132 / **99**	26.48 / **26.56**	10.99 / **6.47**	26.99 / **29.78**	45.1 / **29.7**
20/13	52 / **39**	26.37 / **26.58**	11.19 / **8.28**	26.59 / **28.63**	47.1 / **37.6**
15/10	40 / **30**	26.46 / **26.76**	13.31 / **9.77**	25.71 / **27.88**	54.0 / **43.5**

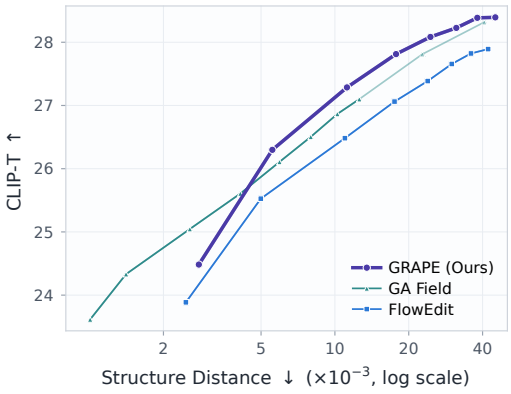


Figure 7: **Component analysis on SD3.** CFG values beyond the range used in the main experiments are shown with reduced opacity.

Table 3: **Editing-field localization.** Accumulated field responses evaluated against the spatial masks of 161 PIE-Bench examples, at matched edit strength with adaptive integration disabled. Higher values indicate that more of the response falls inside the editing region, and the GA field improves all four metrics.

Metric	FlowEdit	GA field
CLIP-T	26.48	26.50
AUROC ↑	0.666	**0.803**
IoU ↑	0.245	**0.404**
Lift ↑	1.164	**1.365**
Mass-in ↑	0.274	**0.293**

the magnitude of each editing field over all steps and evaluate it against the PIE-Bench masks. As reported in Table 3, the GA field concentrates its response more closely within the editing region, improving all four localization metrics.

Reduced inference budgets. Table 2 compares both methods under reduced budgets. Reducing the number of steps does not improve FlowEdit, and GRAPE outperforms it on all metrics at every budget. With 10 editing updates and 30 velocity evaluations, GRAPE also surpasses FlowEdit at its full budget of 33 updates and 132 evaluations, using less than a quarter of the computation.

5 Conclusion

We revisited inversion-free flow editing through the construction and integration of its editing dynamics. Trajectory-derived fields compare the source and target conditions at different noisy states, which lets their velocity difference reflect a denoising response to the state difference in addition to the condition change. GRAPE instead evaluates both conditions at a shared observation of the current edit, yielding a field aligned with the gradient of target-relative preference under an ideal model, and applies it with progress-adaptive integration that lengthens the step when the edit falls behind the noise schedule. Across three flow backbones, GRAPE consistently improves the trade-off between target alignment and source preservation without training or inversion. We hope this perspective provides a useful basis for analyzing and designing inversion-free editing dynamics.

Limitations. Our theoretical characterization holds for the population-optimal velocity, which the pretrained model only approximates. The progress reference relies on a single-step target prediction that can be inaccurate at high noise levels. Extending the dynamics to video editing, where temporal consistency must also be maintained, is left for future work.

AI USE STATEMENT

In this work, we used generative AI tools to assist with refining conceptual frameworks and mathematical formulations, receiving feedback on research methodology and experimental design, implementing and debugging code, interpreting preliminary experimental results, and translating and editing the manuscript. We additionally used generative AI tools for literature search and summarization, paper organization, title suggestions, and readability improvements.

All AI-assisted mathematical arguments and research claims were independently examined by the authors. AI-assisted code was manually reviewed and tested, and all reported experimental results were produced and verified using our evaluation pipeline. The authors take full responsibility for the final content of this work, including all text, claims, code, and artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

We describe the method and provide pseudocode in Appendix C, with implementation details, hyperparameters, evaluation protocols, and additional ablations in Appendix D. We report the pretrained models, dataset, sampling configurations, guidance scales, and metric implementations used in our experiments. An anonymized implementation and instructions for reproducing the main results are included in the supplementary material.

ETHICS STATEMENT

This work studies training-free image editing using publicly available pretrained generative models and an established evaluation benchmark, and involves no new data collection or human subjects. As with other image editing techniques, the proposed method may inherit biases from the underlying models and could potentially be misused to create misleading visual content. We encourage responsible use in accordance with the licenses and usage policies of the corresponding models and datasets.

REFERENCES

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, 2023. URL https://openreview.net/forum?id=li7qeBbCR1t.

Omri Avrahami, Or Patashnik, Ohad Fried, Egor Nemchinov, Kfir Aberman, Dani Lischinski, and Daniel Cohen-Or. Stable flow: Vital layers for training-free image editing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025*, pp. 7877–7888, 2025. doi: 10.1109/CVPR52734.2025.00738.

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*, pp. 18662–18672, 2026.

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

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 *IEEE/CVF International Conference on Computer Vision, ICCV 2023*, pp. 22503–22513, 2023. doi: 10.1109/ICCV51070.2023.02062.

Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*, pp. 11462–11471, 2022. doi: 10.1109/CVPR52688.2022.01118.

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems 34, NeurIPS 2021*, pp. 8780–8794, 2021.

Taichi Endo, Guoqing Hao, and Kazuhiko Sumi. FlowSlider: Training-free continuous image editing via fidelity-steering decomposition. *CoRR*, abs/2604.02088, 2026. doi: 10.48550/ARXIV.2604.02088.

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, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. DreamSim: Learning new dimensions of human visual similarity using synthetic data. In *Advances in Neural Information Processing Systems 36, NeurIPS 2023*, 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 *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=_CDixzkzeyb.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *CoRR*, abs/2207.12598, 2022. doi: 10.48550/ARXIV.2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. Gotta go fast when generating data with score-based models. *CoRR*, abs/2105.14080, 2021.

Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. PnP inversion: Boosting diffusion-based editing with 3 lines of code. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=FoMZ4ljhVw.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems 35, NeurIPS 2022*, 2022.

Jeongsol Kim, Yeobin Hong, Jonghyun Park, and Jong Chul Ye. FlowAlign: Trajectory-regularized, inversion-free flow-based image editing. In *International Conference on Learning Representations*, volume 2026, pp. 121814–121846, 2026.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. FlowEdit: Inversion-free text-based editing using pre-trained flow models. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pp. 19721–19730. IEEE, 2025. doi: 10.1109/ICCV51701.2025.01834.

Guangzhao Li, Yanming Yang, Chenxi Song, Xiaohong Liu, and Chi Zhang. FlowDirector: Training-free flow steering for precise text-to-video editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7805–7815, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems 35, NeurIPS 2022*, 2022.

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 *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=aBsCjcPu_tE.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 6038–6047. IEEE, 2023. doi: 10.1109/CVPR52729.2023.00585.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763, 2021.

Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, 2023. URL https://openreview.net/forum?id=4PJUBT9f2Ol.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 10674–10685. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01042.

Litu Rout, Yujia Chen, Nataniel Ruiz, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Semantic image inversion and editing using rectified stochastic differential equations. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=Hu0FSOSEyS.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Xuan Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. Dual diffusion implicit bridges for image-to-image translation. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, 2023. URL https://openreview.net/forum?id=5HLoTvVGDe.

Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Splicing ViT features for semantic appearance transfer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*, pp. 10738–10747, 2022. doi: 10.1109/CVPR52688.2022.01048.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 1921–1930. IEEE, 2023. doi: 10.1109/CVPR52729.2023.00191.

Bram Wallace, Akash Gokul, and Nikhil Naik. EDICT: Exact diffusion inversion via coupled transformations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 22532–22541. IEEE, 2023. doi: 10.1109/CVPR52729.2023.02158.

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 *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/wang25ce.html.

Sheng-Yu Wang, Yotam Nitzan, Aaron Hertzmann, Jun-Yan Zhu, Eli Shechtman, Alexei A Efros, and Richard Zhang. The many senses of visual similarity: A text-prompted image perceptual metric. *arXiv preprint arXiv:2607.18237*, 2026.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. doi: 10.1109/TIP.2003.819861.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Shengming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-Image technical report. *CoRR*, abs/2508.02324, 2025. doi: 10.48550/ARXIV.2508.02324.

Chenxi Xie, Minghan Li, Shuai Li, Yuhui Wu, Qiaosi Yi, and Lei Zhang. DNAEdit: Direct noise alignment for text-guided rectified flow editing. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/b44ae90136013a8d0e2d24f6015b6097-Abstract-Conference.html.

Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with language-guided diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 9454–9461. IEEE, 2024. doi: 10.1109/CVPR52733.2024.00903.

Zitong Xu, Huiyu Duan, Zhongpeng Ji, Xinyun Zhang, Yutao Liu, Xiongkuo Min, Ke Gu, Jian Zhang, Shusong Xu, Jinwei Chen, Bo Li, and Guangtao Zhai. EditHF-1M: A million-scale rich human preference feedback for image editing. In *Computer Vision – ECCV 2026*, volume 17062 of *Lecture Notes in Computer Science*, pp. 278–298. Springer, 2026. doi: 10.1007/978-3-032-37098-3_16.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

Appendix

Table of Contents

Appendix A Related Work 14

Appendix B Theoretical Analysis 15

Appendix C Algorithm 17

Appendix D Experimental Details 17

Appendix E Additional Quantitative Results 18

Appendix F Additional Qualitative Results 22

A Related Work

Inversion-based editing. Editing a real image with a pretrained model typically begins by recovering a noise sample that reproduces it. DDIM inversion (Song et al., 2021a) gives the basic procedure, and subsequent work reduces the reconstruction error it incurs under guidance (Mokady et al., 2023; Wallace et al., 2023; Ju et al., 2024) or controls generation through attention and feature injection (Hertz et al., 2023; Tumanyan et al., 2023; Cao et al., 2023; Avrahami et al., 2025). The same difficulty reappears for rectified flow models, where inversion is formulated as optimal control (Rout et al., 2025), solved with higher-order solvers (Wang et al., 2025), or avoided by aligning the noise in the forward direction (Xie et al., 2025). All of these require a separate inversion pass, whose cost and residual error motivate the inversion-free setting we study.

Inversion-free editing. Inversion-free methods keep the edited image in clean image space and update it with a direction obtained from conditional predictions. FlowEdit (Kulikov et al., 2025) constructs such a direction for flow models by replacing the source and target trajectories with states built from the source image and the accumulated edit. Subsequent work adds corrections on top of this construction, including trajectory regularization (Kim et al., 2026), continuous control of editing strength (Endo et al., 2026), and an extension to video editing (Li et al., 2026). These methods differ in the corrections they apply, but all of them evaluate the source and target conditions at distinct states, which is the property we revisit in Section 3.1.

Score distillation for editing. A separate line formulates editing as an optimization problem in which the edited image is updated with gradients derived from a pretrained model, using the difference between target- and source-conditioned predictions to suppress the bias that would otherwise degrade the source content. DRFS (Beaudouin et al., 2026) adapts this formulation to rectified flow models with a two-branch objective. These methods optimize the image with repeated gradient steps outside any sampling trajectory, whereas we define a field that is integrated once along the noise schedule, and their two branches are anchored differently rather than evaluated at a single observation of the current edit.

Guidance and sampling schedules. Classifier and classifier-free guidance (Dhariwal & Nichol, 2021; Ho & Salimans, 2022) steer generation with a term proportional to the gradient of an implicit classifier, the interpretation we use in Proposition 1. A related line studies how sampling

steps are distributed along the schedule, either by adapting the step size to the local integration error (Jolicoeur-Martineau et al., 2021; Lu et al., 2022) or by designing the schedule in advance (Karras et al., 2022). Our criterion is different, comparing the edit accumulated in the clean state against the progress the schedule assumes. The uneven accumulation we observe is consistent with the coarse-to-fine behavior reported for diffusion models (Choi et al., 2022; Rissanen et al., 2023).

B Theoretical Analysis

B.1 Gradient alignment of the shared state field

We prove Proposition 1 for $0 < \sigma < 1$ and the Gaussian linear probability path $x_\sigma = (1-\sigma)x_0 + \sigma\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ is independent of x_0 and the condition c. Let $m_c(x, \sigma) = \mathbb{E}[x_0 \mid x_\sigma = x, c]$ and denote the conditional score by $h_c(x, \sigma) = \nabla_x \log p_\sigma(x \mid c)$. Conditioned on x_0, the path gives $x_\sigma \sim \mathcal{N}((1-\sigma)x_0, \sigma^2 I)$. The score of this Gaussian mixture is the posterior average of the component scores,

$$h_c(x, \sigma) = \mathbb{E}\left[-\frac{x - (1-\sigma)x_0}{\sigma^2} \;\middle|\; x_\sigma = x, c \right].$$

Taking the conditional expectation gives

$$h_c(x, \sigma) = -\frac{x - (1-\sigma)m_c(x, \sigma)}{\sigma^2}. \tag{10}$$

The path velocity is $\epsilon - x_0$. Since $\epsilon - x_0 = (x_\sigma - x_0)/\sigma$, conditioning on $x_\sigma = x$ gives

$$v_c^*(x, \sigma) = \mathbb{E}[\epsilon - x_0 \mid x_\sigma = x, c] = \frac{x - m_c(x, \sigma)}{\sigma}. \tag{11}$$

Solving equation 10 for the conditional mean gives $m_c(x, \sigma) = (x + \sigma^2 h_c(x, \sigma))/(1-\sigma)$. Substituting this expression into equation 11 gives

$$v_c^*(x, \sigma) = -\frac{x + \sigma h_c(x, \sigma)}{1 - \sigma}. \tag{12}$$

At a shared evaluation state x, the state term $-x/(1-\sigma)$ is identical for the source and target conditions. Subtracting the two velocities cancels that term and gives

$$v^*_{\mathrm{src}}(x, \sigma) - v^*_{\mathrm{tgt}}(x, \sigma) = \frac{\sigma}{1-\sigma} \nabla_x \log \frac{p_\sigma(x \mid c_{\mathrm{tgt}})}{p_\sigma(x \mid c_{\mathrm{src}})}. \tag{13}$$

For $s_i(z_i) = (1-\sigma_i)z_i + \sigma_i \epsilon_i$, define

$$F_i(z_i; \epsilon_i) = \log \frac{p_{\sigma_i}(s_i(z_i) \mid c_{\mathrm{tgt}})}{p_{\sigma_i}(s_i(z_i) \mid c_{\mathrm{src}})}. \tag{14}$$

Holding ϵ_i and σ_i fixed gives $\partial s_i / \partial z_i = (1-\sigma_i)I$. Applying the chain rule and then equation 13 yields

$$\nabla_{z_i} F_i(z_i; \epsilon_i) = (1-\sigma_i)\big[h_{\mathrm{tgt}}(s_i, \sigma_i) - h_{\mathrm{src}}(s_i, \sigma_i)\big] = \frac{(1-\sigma_i)^2}{\sigma_i} u_i^{\mathrm{GA},*}. \tag{15}$$

This proves Proposition 1. In particular, when $u_i^{\mathrm{GA},*} \neq 0$, its directional derivative is

$$\left\langle \nabla_{z_i} F_i, u_i^{\mathrm{GA},*} \right\rangle = \frac{(1-\sigma_i)^2}{\sigma_i} \left\| u_i^{\mathrm{GA},*} \right\|^2 > 0. \tag{16}$$

The result concerns a local step at a fixed noise level. It does not assert that the preference increases along the entire editing trajectory, where both the state and the noise level change.

Classifier free guidance. The proposition above uses the population-optimal conditional velocities. For standard classifier free guidance, write $\tilde{v}_c = v^*_\varnothing + w_c(v^*_c - v^*_\varnothing)$, where $v^*_\varnothing$ is the ideal unconditional velocity. If the source and target branches use weights w_{src} and w_{tgt}, respectively, expanding the guidance formula gives

$$\tilde{v}_{\mathrm{src}} - \tilde{v}_{\mathrm{tgt}} = w_{\mathrm{src}} v^*_{\mathrm{src}} - w_{\mathrm{tgt}} v^*_{\mathrm{tgt}} + (w_{\mathrm{tgt}} - w_{\mathrm{src}}) v^*_\varnothing.$$

Substituting equation 12 into these three velocities and writing $h_c = \nabla_s \log p_{\sigma_i}(s_i \mid c)$ gives

$$\tilde{v}_{\mathrm{src}} - \tilde{v}_{\mathrm{tgt}} = \frac{\sigma_i}{1-\sigma_i}\big[w_{\mathrm{tgt}} h_{\mathrm{tgt}} - w_{\mathrm{src}} h_{\mathrm{src}} + (w_{\mathrm{src}} - w_{\mathrm{tgt}}) h_{\varnothing}\big]. \tag{17}$$

The state terms cancel because the coefficients of the three velocities sum to zero. Since each score is a gradient of a log density, the bracket is the gradient of the following potential.

$$\begin{aligned}\tilde{F}_i(z_i; \epsilon_i) = w_{\mathrm{tgt}} \log p_{\sigma_i}(s_i \mid c_{\mathrm{tgt}}) - w_{\mathrm{src}} \log p_{\sigma_i}(s_i \mid c_{\mathrm{src}}) \\ + (w_{\mathrm{src}} - w_{\mathrm{tgt}}) \log p_{\sigma_i}(s_i \mid \varnothing).\end{aligned} \tag{18}$$

The same calculation gives

$$\nabla_{z_i} \tilde{F}_i = \frac{(1-\sigma_i)^2}{\sigma_i} \left(\tilde{v}_{\mathrm{src}}(s_i, \sigma_i) - \tilde{v}_{\mathrm{tgt}}(s_i, \sigma_i)\right). \tag{19}$$

The chain rule contributes the factor $1-\sigma_i$ in equation 19. When both branches use the same positive weight w, the unconditional terms cancel and $\tilde{F}_i = wF_i$. With unequal weights, the field aligns with this generalized CFG potential rather than the unweighted target/source log density ratio. A learned model only approximates these ideal velocities.

B.2 The state dependent term in trajectory based fields

The ideal FlowEdit field evaluates the source and target velocities at q_i and p_i, respectively. Adding and subtracting the source velocity at p_i gives

$$\begin{aligned}u_i^{\mathrm{FE},*} &= v^*_{\mathrm{src}}(q_i, \sigma_i) - v^*_{\mathrm{tgt}}(p_i, \sigma_i) \\ &= \left[v^*_{\mathrm{src}}(p_i, \sigma_i) - v^*_{\mathrm{tgt}}(p_i, \sigma_i)\right] \\ &\quad + \left[v^*_{\mathrm{src}}(q_i, \sigma_i) - v^*_{\mathrm{src}}(p_i, \sigma_i)\right].\end{aligned} \tag{20}$$

Equation equation 13 identifies the first term as proportional to a target/source density ratio gradient at p_i for ideal conditional velocities. The same interpretation holds up to a common factor when both branches use equal CFG weights. The second term compares the same source condition at two different states. Since $p_i - q_i = z_i - x_{\mathrm{src}}$, it depends on the edit displacement even when the condition is fixed. To express this state change explicitly, let $\gamma_i(t) = p_i + t(q_i - p_i)$ for $t \in [0, 1]$. If the source velocity is differentiable, the chain rule and the fundamental theorem of calculus give

$$\begin{aligned}&v^*_{\mathrm{src}}(q_i, \sigma_i) - v^*_{\mathrm{src}}(p_i, \sigma_i) \\ &\quad = \int_0^1 \frac{d}{dt} v^*_{\mathrm{src}}(\gamma_i(t), \sigma_i)\, dt \\ &\quad = \int_0^1 J_x v^*_{\mathrm{src}}(\gamma_i(t), \sigma_i)(q_i - p_i)\, dt.\end{aligned} \tag{21}$$

If this velocity is L_i-Lipschitz on the segment joining p_i and q_i, then

$$\|v^*_{\mathrm{src}}(q_i, \sigma_i) - v^*_{\mathrm{src}}(p_i, \sigma_i)\| \le L_i \|q_i - p_i\| = L_i \|z_i - x_{\mathrm{src}}\|. \tag{22}$$

This generic bound describes only the magnitude of the state dependent term. The decomposition does not imply that the term helps or hurts editing or source preservation. Their net effects are empirical questions addressed in the main text.

B.3 Interpreting the clean-state progress reference

To interpret the reference used by adaptive integration, consider an oracle target endpoint x_{tgt} and let $d^* = x_{\mathrm{tgt}} - x_{\mathrm{src}}$. The linear probability path motivates the clean-state reference

$$a^*(\sigma) = x_{\mathrm{src}} + (1-\sigma) d^*. \tag{23}$$

At the end of step i, this reference is $a^*(\sigma_{i+1})$. It measures progress toward a fixed target endpoint, but is not generally a trajectory of the editing ODE, whose marginal velocity field need not produce a straight path. If editing starts at $\sigma_0 < 1$, the reference may also begin ahead of $z_0 = x_{\mathrm{src}}$.

For $d^* \neq 0$, define the realized progress as the projection of the clean editing state onto d^*, and compare it with the progress indexed by the noise schedule:

$$r_i = \frac{\langle z_i - x_{\mathrm{src}}, d^* \rangle}{\|d^*\|^2}, \qquad r_i^* = 1 - \sigma_i, \qquad L_i^* = r_i^* - r_i. \tag{24}$$

Under a nominal update $z_{i+1} = z_i + \ell_i u_i$, where $\ell_i = \sigma_i - \sigma_{i+1}$, the schedule advances by ℓ_i, whereas realized progress advances by $\ell_i \langle u_i, d^* \rangle / \|d^*\|^2$. Thus,

$$L_{i+1}^* - L_i^* = \ell_i \left(1 - \frac{\langle u_i, d^* \rangle}{\|d^*\|^2} \right). \tag{25}$$

At the terminal level $\sigma_n = 0$, the reference reaches x_{tgt}, giving

$$L_n^* = \frac{\langle x_{\mathrm{tgt}} - z_n, d^* \rangle}{\|d^*\|^2}. \tag{26}$$

A positive terminal lag means that the edit has not completed the displacement along d^*. A positive intermediate lag alone does not imply a positive terminal lag, since later updates may recover the difference.

In practice, the target endpoint is unknown. We therefore estimate it from the target velocity at the current noisy observation s_i:

$$\hat{x}_{\mathrm{tgt},i} = s_i - \sigma_i v_{\mathrm{tgt}}(s_i, \sigma_i), \qquad d_i = \hat{x}_{\mathrm{tgt},i} - x_{\mathrm{src}}, \qquad a_{i+1} = x_{\mathrm{src}} + (1 - \sigma_{i+1}) d_i. \tag{27}$$

For the ideal unguided velocity, equation 11 identifies this clean prediction with $\mathbb{E}[x_0 \mid x_{\sigma_i} = s_i, c_{\mathrm{tgt}}]$. With guidance or an approximate network, this equality need not hold. The resulting a_{i+1} is therefore a comparison reference indexed by the next noise level, not a prediction of the editing trajectory.

C Algorithm

Algorithm 1 summarizes one editing run, where $\widetilde{v}_c$ denotes the guidance-applied velocity. The unconditional prediction is shared across the two conditions, so each step uses three model evaluations on backbones with explicit classifier-free guidance.

Algorithm 1 GRAPE: Gradient-Aligned, Progress-adaptive Editing

Require: x_{src}, prompts $c_{\mathrm{src}}, c_{\mathrm{tgt}}$, schedule $\sigma_0 > \cdots > \sigma_N = 0$, cap M, constant ε

$z_0 \leftarrow x_{\mathrm{src}}$
for $i = 0$ **to** $N - 1$ **do**
  $s_i \leftarrow (1 - \sigma_i) z_i + \sigma_i \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, I)$ ▷ shared observation
  $u_i \leftarrow \widetilde{v}_{\mathrm{tgt}}(s_i, \sigma_i) - \widetilde{v}_{\mathrm{src}}(s_i, \sigma_i)$ ▷ GA field
  $\ell_i \leftarrow \sigma_i - \sigma_{i+1}, \quad z_{i+1}^{\mathrm{raw}} \leftarrow z_i + \ell_i u_i$
  $d_i \leftarrow s_i - \sigma_i \widetilde{v}_{\mathrm{tgt}}(s_i, \sigma_i) - x_{\mathrm{src}}, \quad a_{i+1} \leftarrow x_{\mathrm{src}} + (1 - \sigma_{i+1}) d_i$ ▷ reference
  $g_i \leftarrow \langle a_{i+1} - z_{i+1}^{\mathrm{raw}}, d_i \rangle, \quad c_i \leftarrow \langle u_i, d_i \rangle$
  $\Delta \ell_i \leftarrow g_i / (c_i + \varepsilon)$ **if** $g_i > 0$ **and** $c_i > \varepsilon$, **else** 0
  $z_{i+1} \leftarrow z_i + \min(\ell_i + \Delta \ell_i,\, M \ell_i)\, u_i$
end for
return z_N

D Experimental Details

Implementation and benchmark. Our implementation builds on the public FlowAlign repository (Kim et al., 2026), which follows the FlowEdit reference implementation (Kulikov et al., 2025). We retain its sampler structure, prompt-embedding handling, and PIE-Bench data loader, and implement our editors through the same interface as the baselines. We evaluate all 700 PIE-Bench images (Ju et al., 2024), covering ten editing categories. Images and edit masks are processed at 512×512 resolution. We use all samples and perform no per-sample tuning. Standard metrics are computed using the unmodified PnPInversion evaluator

(`PnPInversion/evaluation/matrics_calculator.py`), which scores CLIP-T with CLIP ViT-L/14 (`openai/clip-vit-large-patch14`) (Radford et al., 2021) and Structure Distance with DINO ViT-B/8 layer-11 key self-similarity (Tumanyan et al., 2022).

Backbones and evaluation budgets. We evaluate SD3 Medium (Esser et al., 2024), FLUX.1-dev (Black Forest Labs, 2024), and Qwen-Image (Wu et al., 2025). SD3 Medium and Qwen-Image use explicit classifier-free guidance, whereas FLUX.1-dev is guidance-distilled and receives guidance as a conditioning input. For FLUX.1-dev, we cycle the latent-grid phase across steps and apply the same procedure to all compared methods.

Inversion-based methods require both inversion and editing passes, while inversion-free methods start partway through the schedule, so we match the number of active steps along the trajectory rather than the nominal schedule length. For SD3 Medium and Qwen-Image, FlowEdit, SDEdit, DNAEdit, and GRAPE use steps 33–50 of a $T = 50$ schedule, FlowAlign runs 33 steps from $\sigma = 1$, and DDIB uses 17 inversion steps followed by 17 editing steps. For FLUX.1-dev, inversion-free methods use steps 24–28 of a $T = 28$ schedule, FlowAlign runs 24 steps, and DDIB uses 12 inversion steps followed by 12 editing steps. DRFS is the one method whose official configuration spans the full schedule ($n_{\text{start}} = T$): we hold it to the shared budget on SD3 Medium (steps 33–50) and FLUX.1-dev (steps 24–28), and leave it at its full $T = 50$ schedule on Qwen-Image, where it therefore takes 50 steps against the 33 of every other method.

Source guidance is 3.5 for SD3 Medium and 1.5 for FLUX.1-dev and Qwen-Image. For every method, target guidance is swept over $\{2.5, 5.0, 7.5, 10.0, 12.5, 15.0, 17.5, 20.0\}$ on SD3 Medium, $\{1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0\}$ on FLUX.1-dev, and $\{2, 4, 6, 8, 10, 12, 14\}$ on Qwen-Image. Trade-off curves use the full sweep. For single-point comparisons, we select the configuration whose CLIP-T is closest to a fixed per-backbone anchor. The anchor is 26.5 for SD3 Medium and Qwen-Image, and 25.75 for FLUX.1-dev, the highest CLIP-T the inversion-free baselines attain on that backbone. Methods that do not reach the anchor are reported at their closest configuration and marked. Such points should not be interpreted as preservation wins, since a method that makes little change can trivially preserve the source.

E Additional Quantitative Results

E.1 Trade-offs with Additional Preservation Metrics

We complement Structure Distance with PSNR, LPIPS, MSE, and SSIM, computed over regions outside the PIE-Bench edit masks. For each backbone, we plot these preservation metrics against CLIP-T across the target-guidance sweep. Higher PSNR and SSIM, and lower LPIPS and MSE, indicate better preservation. The trade-off trends are broadly consistent with the main results, although the relative gaps vary across metrics and backbones. This agreement suggests that our findings are not specific to Structure Distance.

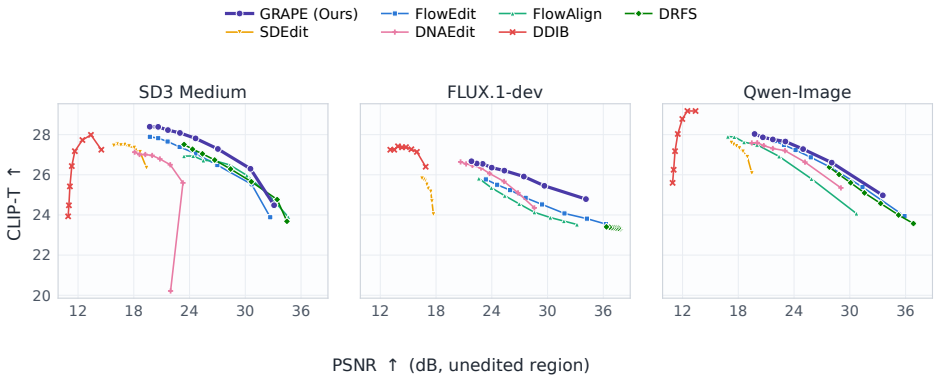


Figure 8: Trade-off between CLIP-T and PSNR.

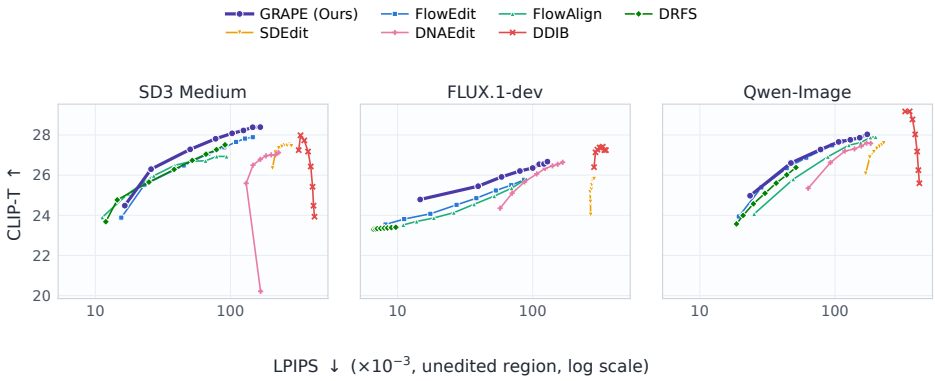


Figure 9: Trade-off between CLIP-T and LPIPS.

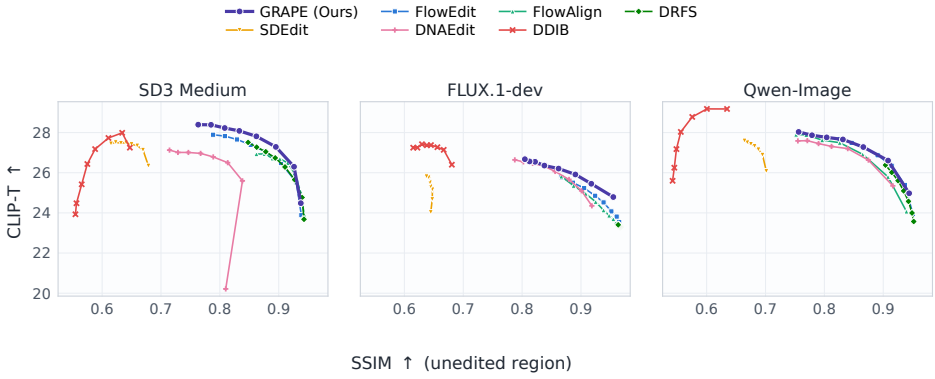


Figure 10: Trade-off between CLIP-T and SSIM.

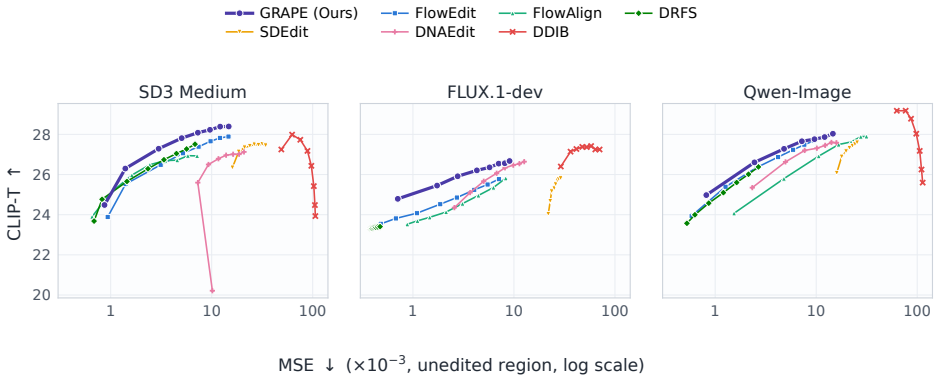


Figure 11: Trade-off between CLIP-T and MSE.

E.2 Perceptual and Human-Aligned Metrics Across Backbones

Tables 4 and 5 report DreamSim and EditHF at the selected operating points on FLUX.1-dev and Qwen-Image. On both backbones, GRAPE attains the lowest Structure Distance and the highest EditHF preservation and visual-quality scores, with EditHF alignment close to the best method. On FLUX.1-dev it also has the lowest DreamSim, while on Qwen-Image its DreamSim is comparable to FlowEdit and DRFS. The highest alignment scores are obtained by FlowAlign, which on Qwen-Image does not reach the anchor and shows substantially larger structural changes. DRFS is omitted

on FLUX.1-dev, as none of its configurations could be matched to the anchor on this guidance-distilled backbone.

Table 4: **Perceptual and human-aligned evaluation on FLUX.1-dev.** Each method uses the configuration closest to the anchor CLIP-T of 25.75. Structure Distance is reported in units of 10^{-3}. EHF-A, EHF-P, and EHF-V denote EditHF alignment, preservation, and quality scores (Xu et al., 2026). The entry marked with * does not reach the anchor within the evaluated guidance range and is reported at its closest configuration.

Method	CLIP-T	Structure ↓	DreamSim ↓	EHF-A ↑	EHF-P ↑	EHF-V ↑
SDEdit	25.77	80.17	0.3605	47.71	36.32	36.93
DDIB*	26.40	94.40	0.4864	47.58	35.84	35.39
FlowEdit	25.77	21.76	0.1718	52.07	48.69	48.16
DNAEdit	25.67	18.02	0.1882	51.70	46.29	45.87
FlowAlign	25.82	24.74	0.1836	**52.38**	47.91	47.34
GRAPE (Ours)	25.91	**9.30**	**0.0972**	52.14	**54.41**	**51.41**

Table 5: **Perceptual and human-aligned evaluation on Qwen-Image.** Each method uses the configuration closest to the anchor CLIP-T of 26.5. Structure Distance is reported in units of 10^{-3}. Entries marked with * do not reach the anchor within the evaluated guidance range and are reported at their closest configuration.

Method	CLIP-T	Structure ↓	DreamSim ↓	EHF-A ↑	EHF-P ↑	EHF-V ↑
SDEdit*	26.89	45.18	0.2539	51.35	42.57	43.81
DDIB	26.25	261.91	0.6292	46.11	35.54	32.52
FlowEdit	26.35	7.65	0.0962	52.49	54.08	51.56
DNAEdit	26.63	17.01	0.1634	52.69	48.69	48.33
FlowAlign*	26.92	30.05	0.2513	**53.01**	48.72	47.59
DRFS	26.38	7.51	**0.0906**	52.59	54.37	51.73
GRAPE (Ours)	26.61	**7.27**	0.0965	52.84	**54.58**	**51.80**

We also report TPIPS (Wang et al., 2026) over the full image (TPIPS-a) and over the unedited regions (TPIPS-bg), where lower is better (Tables 6, 7, and 8). GRAPE attains the lowest TPIPS-a on SD3 Medium and FLUX.1-dev and the lowest TPIPS-bg on FLUX.1-dev, while DRFS is marginally lower in TPIPS-bg on SD3 Medium. On Qwen-Image, FlowEdit and DRFS obtain slightly lower values than GRAPE, with small differences. TPIPS thus supports the preservation trends on SD3 Medium and FLUX.1-dev, without a uniform advantage across backbones.

Table 6: **TPIPS evaluation on SD3 Medium.** Each method uses the configuration closest to the anchor CLIP-T of 26.5. TPIPS-a is measured over the full image and TPIPS-bg over the unedited regions, and lower is better.

Method	CLIP-T	TPIPS-a ↓	TPIPS-bg ↓
SDEdit	26.35	0.2795	0.2110
DDIB	26.43	0.4925	0.3564
FlowEdit	26.48	0.1557	0.0777
DNAEdit	26.50	0.2793	0.1821
FlowAlign	26.49	0.1814	0.0969
DRFS	26.29	0.1316	**0.0665**
GRAPE (Ours)	26.56	**0.1295**	0.0668

E.3 Reverse-Edit Evaluation

We evaluate round-trip consistency on all 700 images by editing each result back toward its source prompt and comparing the reconstruction with the original source (Tables 9, 10, and 11). Forward CLIP-T indicates the strength of the initial edit. GRAPE attains the lowest Structure Distance on all three backbones and the best scores on all four metrics on FLUX.1-dev. On SD3 Medium it also has the lowest LPIPS, while FlowAlign is slightly higher in PSNR and SSIM, and on Qwen-Image it has the highest PSNR, while FlowEdit has slightly better LPIPS and SSIM.

Table 7: **TPIPS evaluation on FLUX.1-dev.** Each method uses the configuration closest to the anchor CLIP-T of 25.75. The entry marked with * does not reach the anchor within the evaluated guidance range and is reported at its closest configuration. DRFS is omitted on this backbone, as none of its configurations could be matched to the anchor.

Method	CLIP-T	TPIPS-a ↓	TPIPS-bg ↓
SDEdit	25.77	0.3339	0.2357
DDIB†	26.40	0.4293	0.2869
FlowEdit	25.77	0.1842	0.0984
DNAEdit	25.67	0.2060	0.1217
FlowAlign	25.82	0.2012	0.1116
GRAPE (Ours)	25.91	**0.1128**	**0.0644**

Table 8: **TPIPS evaluation on Qwen-Image.** Each method uses the configuration closest to the anchor CLIP-T of 26.5. Entries marked with * do not reach the anchor within the evaluated guidance range and are reported at their closest configuration.

Method	CLIP-T	TPIPS-a ↓	TPIPS-bg ↓
SDEdit†	26.89	0.2463	0.1568
DDIB	26.25	0.5303	0.3883
FlowEdit	26.35	0.1230	**0.0661**
DNAEdit	26.63	0.1900	0.1164
FlowAlign†	26.92	0.2564	0.1487
DRFS	26.38	**0.1203**	0.0670
GRAPE (Ours)	26.61	0.1250	0.0682

Table 9: **Reverse-edit reconstruction on SD3 Medium.** Each image is edited with the target prompt and then edited back with the source prompt, and the result is compared with the source image. Forward CLIP-T reports the alignment of the intermediate edit, measured at the configuration closest to the anchor CLIP-T of 26.5. Structure Distance is reported in units of 10^{-3}.

Method	Forward CLIP-T	Structure ↓	PSNR ↑	LPIPS ↓	SSIM ↑
SDEdit	26.35	111.16	12.06	0.4905	0.3714
DDIB	26.43	278.84	7.85	0.6649	0.3183
FlowEdit	26.48	17.34	20.90	0.1286	0.7774
DNAEdit	26.50	45.42	17.35	0.2917	0.6371
FlowAlign	26.49	11.93	**24.15**	0.0941	**0.8235**
DRFS	26.29	13.63	21.91	0.1144	0.7873
GRAPE (Ours)	26.56	**8.97**	23.75	**0.0869**	0.8169

Table 10: **Reverse-edit reconstruction on FLUX.1-dev.** Forward CLIP-T is measured at the configuration closest to the anchor CLIP-T of 25.75, and Structure Distance is reported in units of 10^{-3}. Entries marked with * do not reach the anchor within the evaluated guidance range and are reported at their closest configuration. DRFS is omitted on this backbone, as none of its configurations could be matched to the anchor.

Method	Forward CLIP-T	Structure ↓	PSNR ↑	LPIPS ↓	SSIM ↑
SDEdit	25.82	111.29	11.79	0.5133	0.3985
DDIB†	26.40	164.13	13.05	0.6096	0.4294
FlowEdit	25.77	37.57	17.02	0.2297	0.6811
DNAEdit†	26.47	40.26	17.42	0.2616	0.6558
FlowAlign	25.82	40.96	16.79	0.2266	0.6965
GRAPE (Ours)	25.91	**19.27**	**20.64**	**0.1752**	**0.7144**

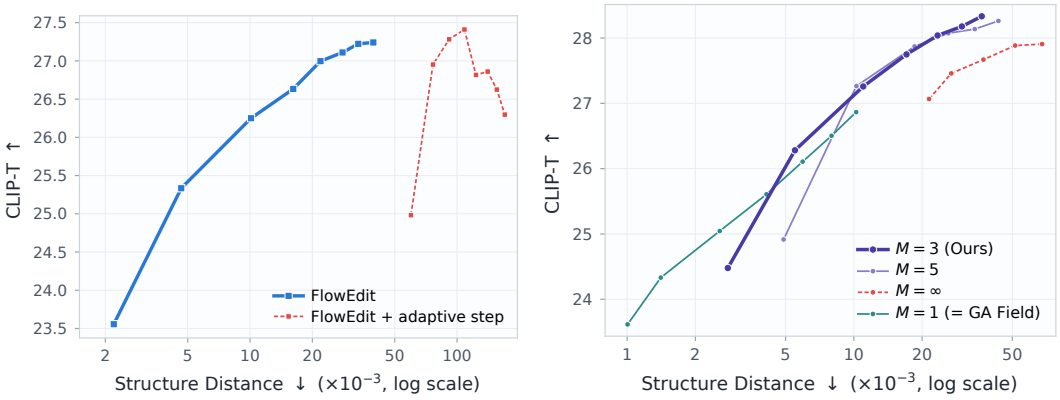
E.4 Further Analysis of Adaptive Integration

We first apply the same adaptive integration rule to the trajectory-derived field of FlowEdit. Figure 12(a) shows that the resulting curve moves toward substantially larger Structure Distance, and

Table 11: **Reverse-edit reconstruction on Qwen-Image.** Forward CLIP-T is measured at the configuration closest to the anchor CLIP-T of 26.5, and Structure Distance is reported in units of 10^{-3}. Entries marked with * do not reach the anchor within the evaluated guidance range and are reported at their closest configuration.

Method	Forward CLIP-T	Structure ↓	PSNR ↑	LPIPS ↓	SSIM ↑
SDEdit†	26.89	79.50	12.84	0.4027	0.4387
DDIB	26.25	271.45	8.20	0.6677	0.3228
FlowEdit	26.35	13.22	21.37	**0.1287**	**0.7960**
DNAEdit	26.63	23.58	19.88	0.2124	0.7413
FlowAlign†	26.92	35.27	18.57	0.2032	0.7405
DRFS	26.38	14.15	21.17	0.1504	0.7685
GRAPE (Ours)	26.61	**12.35**	**21.62**	0.1380	0.7826

that at comparable CLIP-T FlowEdit without adaptive integration preserves the source structure better. The rule controls how far the chosen direction is applied, and does not by itself ensure that a longer step preserves source structure. Extending the application length is therefore effective when the direction reflects the condition change, which the GA field provides by construction.



(a) Adaptive integration applied to FlowEdit

(b) Maximum step multiplier with the GA field

Figure 12: **Adaptive integration on SD3 Medium.** Each curve plots CLIP-T against Structure Distance across the guidance sweep. (a) FlowEdit with and without adaptive integration ($M = 3$). (b) The GA field with adaptive integration under different maximum step multipliers M.

Figure 12(b) examines the maximum step multiplier M when adaptive integration is used with the GA field. Setting $M = 1$ recovers the GA field with standard steps. Increasing the bound to $M = 3$ extends the curve toward stronger target alignment, while $M = 5$ gives a similar trade-off over much of the overlapping range. Removing the bound produces substantially larger structural changes without a corresponding improvement in CLIP-T. These results support using a finite bound in this setting but do not imply that $M = 3$ is optimal for every backbone or step budget.

F Additional Qualitative Results

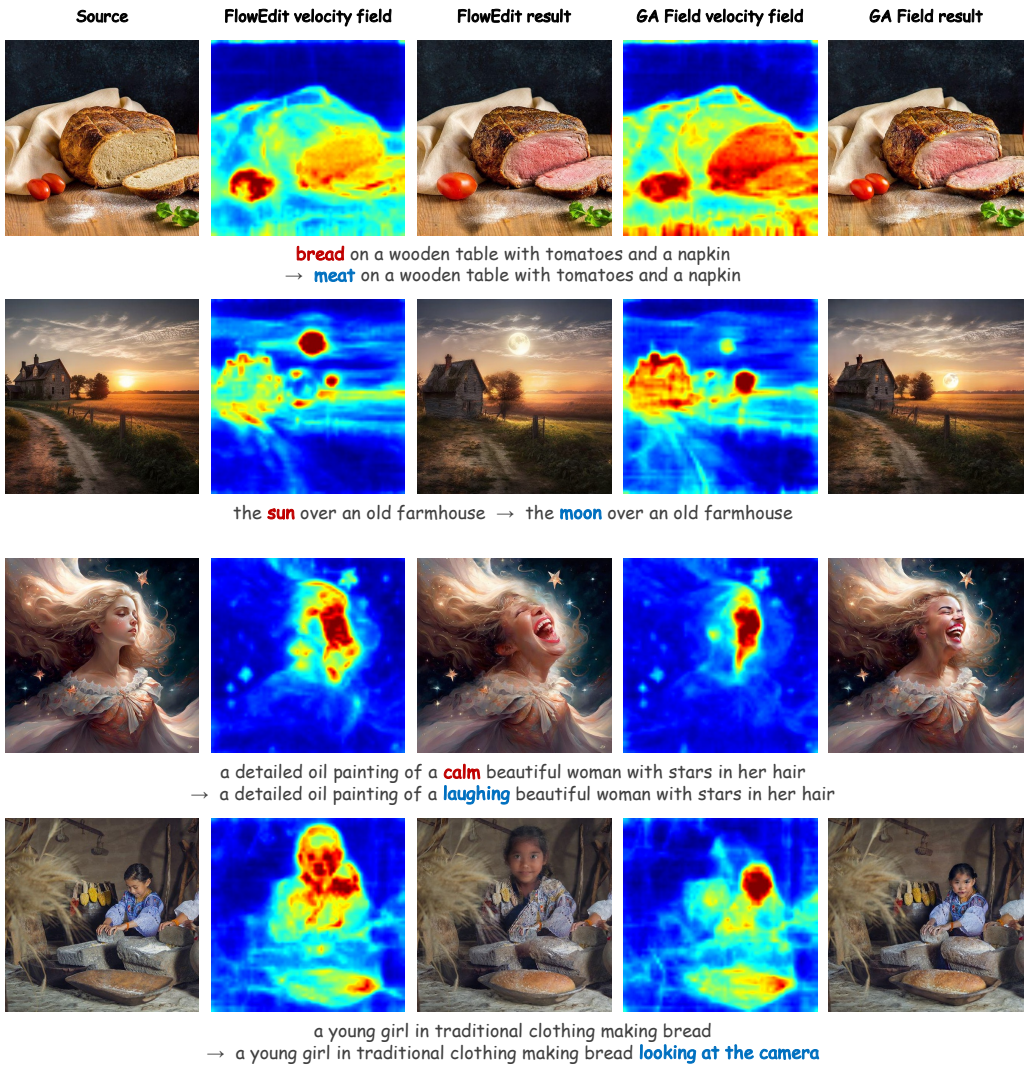


Figure 13: **Qualitative visualization of editing-field responses.** For each example, we show the source image, the FlowEdit response map and result, and the GA-field response map and result. The maps illustrate where each field responds to the requested edit. In these examples, the GA field more selectively emphasizes edit-relevant regions, while its outputs retain more of the surrounding source content.

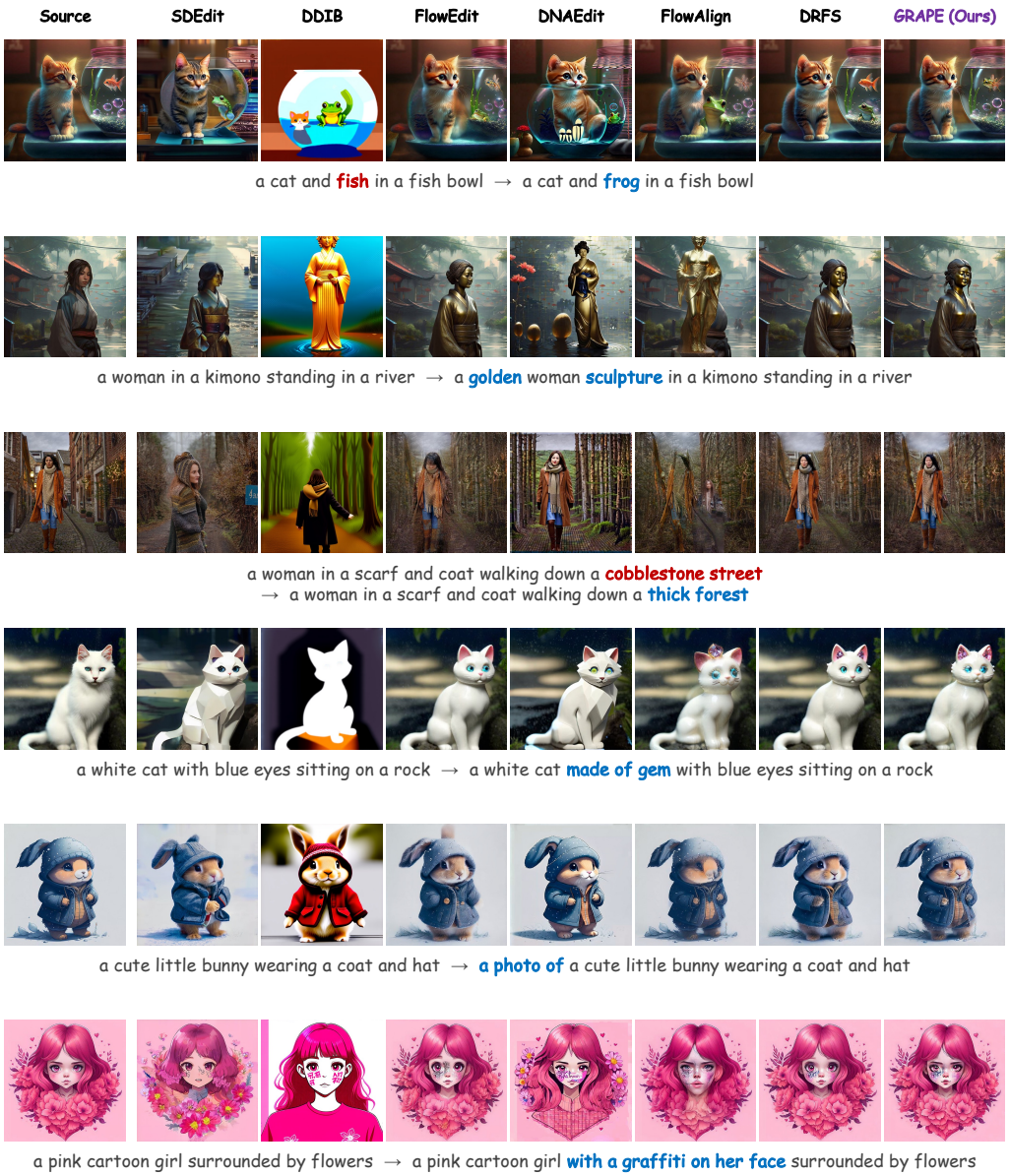


Figure 14: **Additional qualitative results on SD3 Medium.** Columns show the source image and outputs from SDEdit, DDIB, FlowEdit, DNAEdit, FlowAlign, DRFS, and GRAPE, in order.

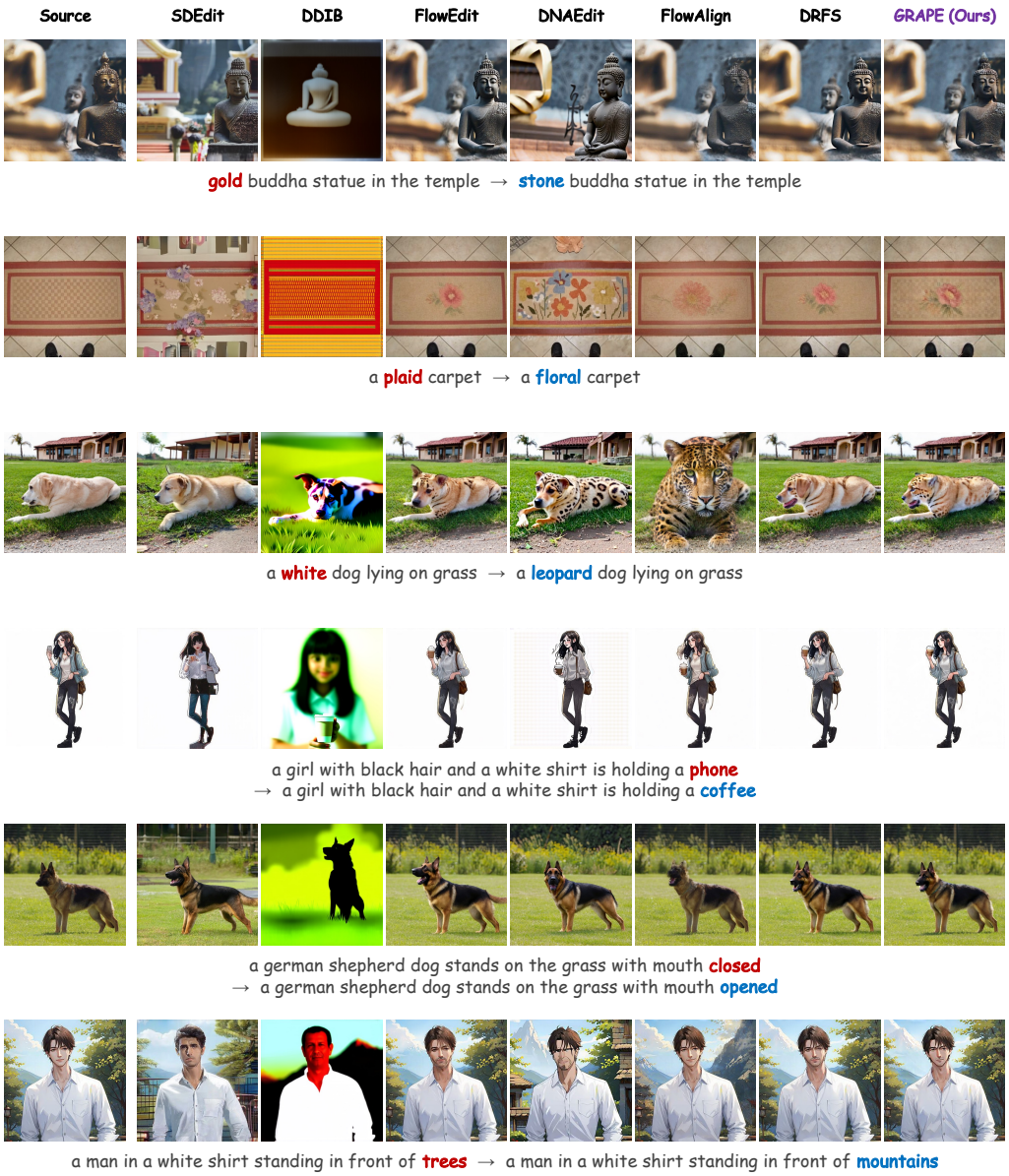


Figure 15: **Additional qualitative results on SD3 Medium.** Columns show the source image and outputs from SDEdit, DDIB, FlowEdit, DNAEdit, FlowAlign, DRFS, and GRAPE, in order.

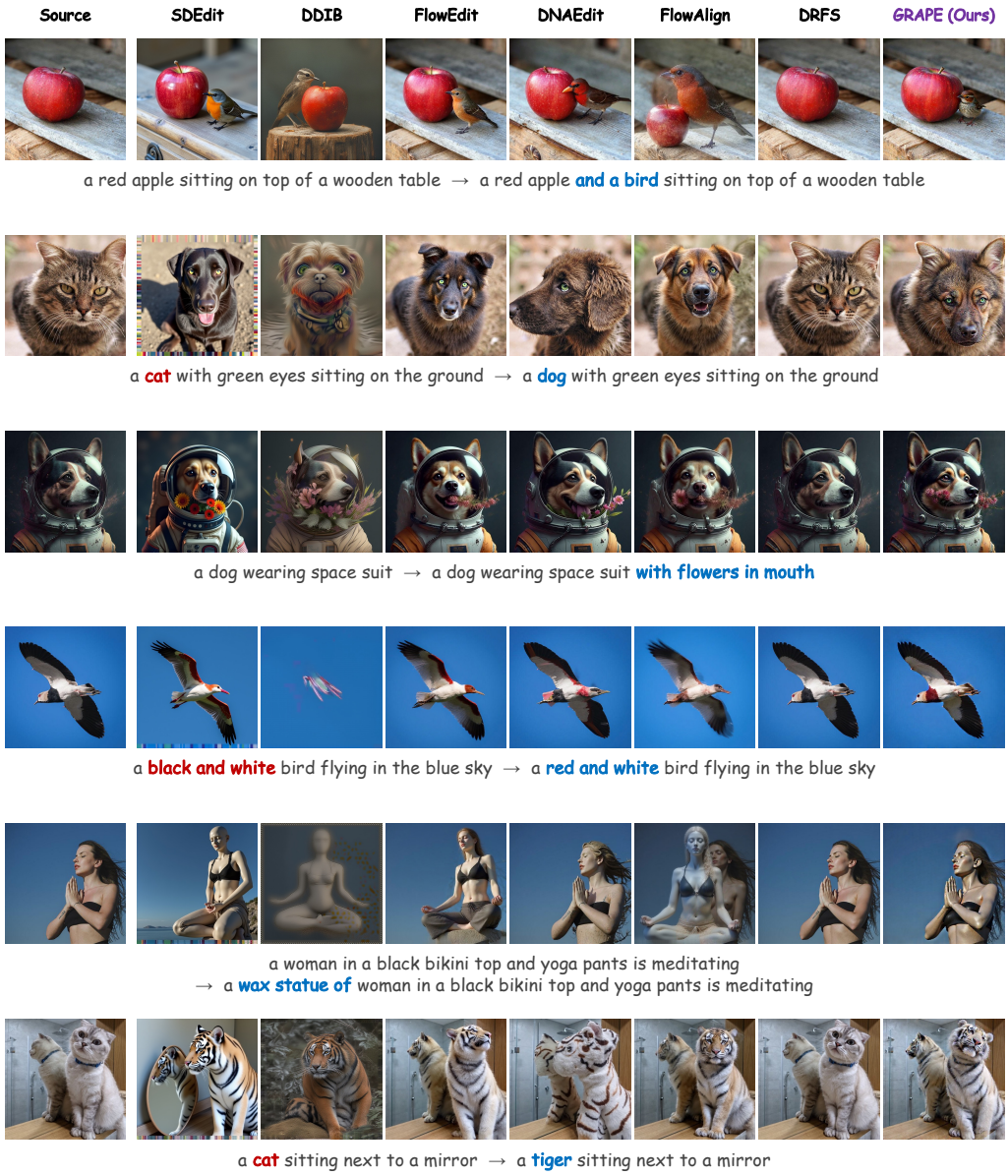


Figure 16: **Additional qualitative results on FLUX.1-dev.** Columns show the source image and outputs from SDEdit, DDIB, FlowEdit, DNAEdit, FlowAlign, DRFS, and GRAPE, in order.

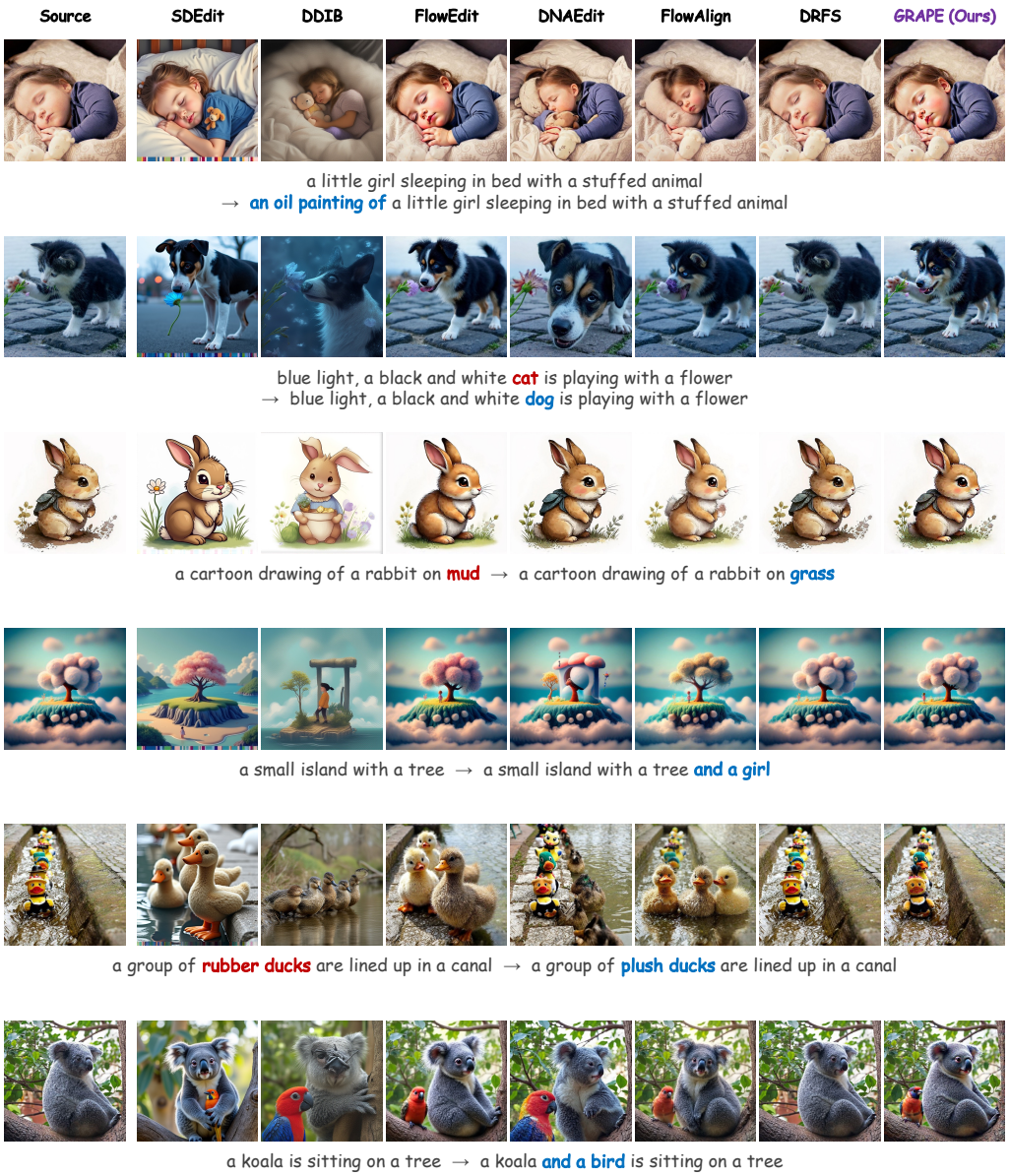


Figure 17: **Additional qualitative results on FLUX.1-dev.** Columns show the source image and outputs from SDEdit, DDIB, FlowEdit, DNAEdit, FlowAlign, DRFS, and GRAPE, in order.

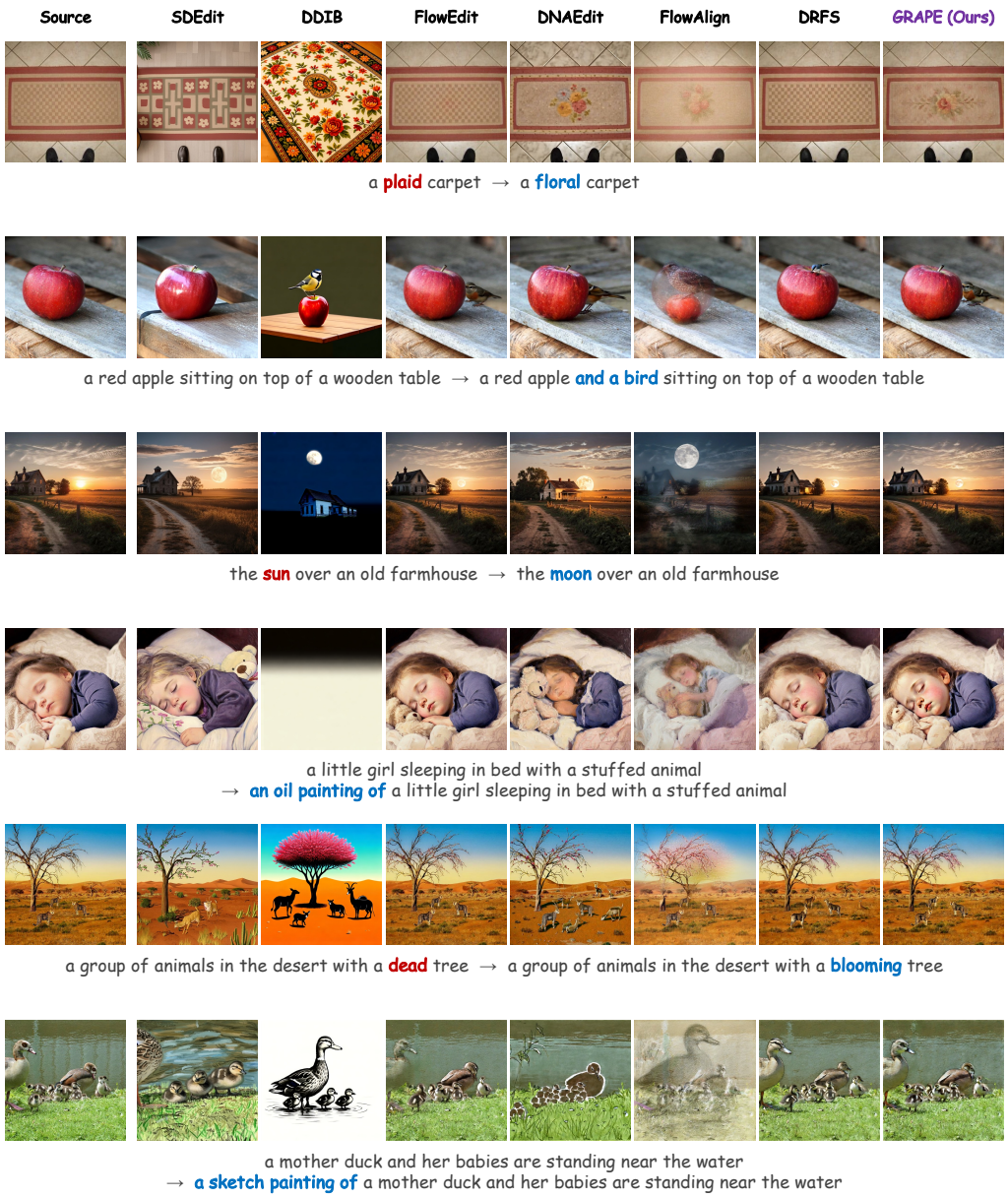


Figure 18: **Additional qualitative results on Qwen-Image.** Columns show the source image and outputs from SDEdit, DDIB, FlowEdit, DNAEdit, FlowAlign, DRFS, and GRAPE, in order.

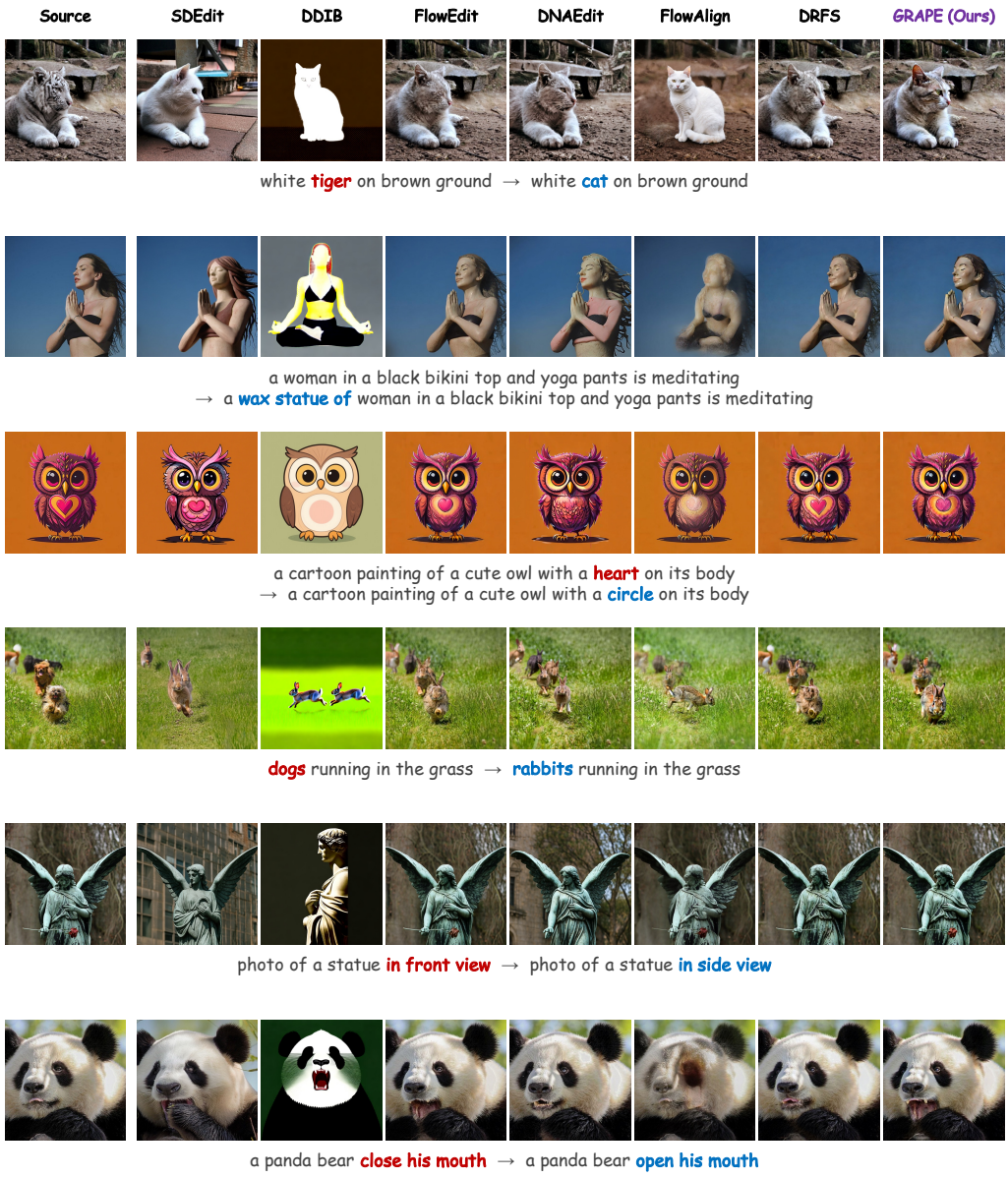


Figure 19: **Additional qualitative results on Qwen-Image.** Columns show the source image and outputs from SDEdit, DDIB, FlowEdit, DNAEdit, FlowAlign, DRFS, and GRAPE, in order.