ONE EVALUATION IS ENOUGH: INVERSION-FREE IMAGE EDITING WITH TRAJECTORY CORRECTION

Anonymous authors
Paper under double-blind review

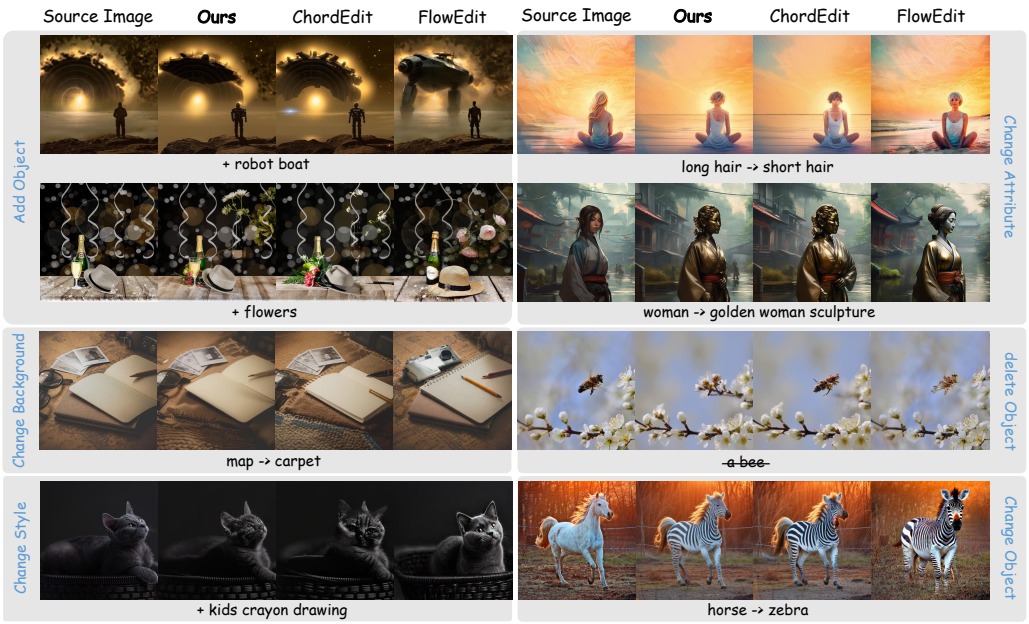


Figure 1: Edits with ECT (ours), ChordEdit, and FlowEdit, covering object addition and removal, attribute and background changes, stylization, and object replacement.

ABSTRACT

Real-image editing commonly relies on inversion and iterative sampling, while efficient one-step editors often stabilize a large update by estimating and averaging editing fields at multiple noise levels. We ask a stricter question: is a single temporal field evaluation sufficient for reliable inversion-free editing? Our answer rests on a previously underused duality in a variance-preserving (VP) one-step model: one source-target query at a shared noisy state simultaneously determines a local angular semantic tangent and a clean endpoint displacement. We integrate the tangent over the analytically known remaining VP angle to obtain a nominal one-step transport, and use the endpoint displacement to correct its long-horizon extrapolation through a constrained projection. This transport-first, correction-second construction yields Endpoint-Calibrated Transport (ECT), a closed-form, training-free editor requiring only one noise level, two text conditions packed into one denoiser batch, and one latent update. ECT requires no image inversion, trajectory integration, second temporal observation, or iterative optimization, and can be directly applied to pretrained epsilon-prediction VP one-step generators. Across the reported benchmark settings, ECT delivers stable gains in editing quality within an extremely short editing time.

1 INTRODUCTION

Text-guided real-image editing must accurately realize a target instruction while preserving unedited regions, spatial layout, and subject identity. Existing approaches use diffusion inversion, feature or attention injection, and iterative denoising to retain source structure. Although effective, inversion optimization and long sampling trajectories incur substantial latency and memory cost, limiting interactive use (Hertz et al., 2022; Mokady et al., 2023; Tumanyan et al., 2023; Cao et al., 2023).

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.

Recent work improves efficiency along two complementary directions. Inversion-free editors construct a direct or consistency-based process between source and target conditions, while distilled generators compress synthesis into one or a few steps (Xu et al., 2024; Kulikov et al., 2025; Liu et al., 2023; Sauer et al., 2024; Dao et al., 2024). Yet high-fidelity direct editors generally still integrate or correct model fields over multiple times. Even when the final edit uses one latent update, representative one-step methods stabilize that update with information collected at more than one noise level (Lu et al., 2026). Reliable editing from one temporal support thus remains challenging.

This paper studies *single-evaluation image editing*: can a stable inversion-free edit be constructed by querying the source and target conditions at the same noisy state and at only one noise level? Multiple condition-time branches can be packed into a single batch, but every additional temporal observation still enlarges the batch, activation traffic, CPU scheduling, and memory pressure. We therefore characterize an editor by its temporal supports, condition-time branches, batched denoiser calls, and latent updates. We seek stable edits without additional temporal observations.

Our key observation is that a single conditional query of a VP one-step model determines one *edit axis*: the angular tangent and clean-endpoint displacement derived from the shared state are collinear. We calibrate its magnitude by integrating the frozen local tangent over the known remaining VP angle, obtaining a nominal *single-evaluation angular transport*. This schedule-based rescaling is a first-order approximation, not a guarantee that the original displacement underestimates the true edit. A closed-form endpoint-exposure shrinkage then moderates the extrapolation along the same axis, yielding *Endpoint-Calibrated Transport* (ECT). The editor needs one noise level, two conditional branches in one denoiser batch, and one latent update, without training, inversion, a second temporal observation, or iterative optimization.

We validate ECT on two real-image editing benchmarks and two representative one-step backbones. The results support using one shared-state evaluation for both semantic transport and endpoint calibration, yielding competitive edits with lower overhead.

Our contributions are threefold:

- **Single-evaluation formulation.** We formulate inversion-free editing under a strict single-temporal-support constraint: the source and target conditions are evaluated at one shared noisy state and followed by one latent update. It separates temporal support from batching and exposes the activation, memory, and scheduling costs of additional branches.
- **Transport-and-correction theory.** We derive a schedule-based nominal transport with a Lipschitz quadrature bound and a closed-form endpoint-exposure projection on the same edit axis. A shrinkage-risk analysis motivates correction under explicit assumptions.
- **Training-free core setting.** ECT operates directly on pretrained epsilon-prediction VP one-step backbones using one noise level, two text branches packed in one denoiser batch, and one latent update, without training, inversion, ODE integration, or iterative optimization. The pipeline combines competitive editing quality with lower execution overhead.

2 Related Work

Diffusion-based real-image editing. Diffusion probabilistic models and their deterministic samplers (Ho et al., 2020; Song et al., 2020), together with classifier-free guidance (Ho & Salimans, 2022), underlie most text-guided editors. Noise-and-denoise editing such as SDEdit (Meng et al., 2022) and instruction-tuned editors such as InstructPix2Pix (Brooks et al., 2023) modify images directly in pixel or latent space, while text-guided diffusion editors manipulate cross-attention, inject intermediate features, or share self-attention to transfer target semantics while retaining source layout (Hertz et al., 2022; Tumanyan et al., 2023; Cao et al., 2023). Their preservation ability commonly depends on DDIM inversion, null-text optimization, or source-reference branches that reconstruct an input-specific diffusion trajectory (Mokady et al., 2023; Miyake et al., 2025). Although these mechanisms provide fine-grained control, the reconstruction process introduces many sequential denoising steps, per-image optimization, and storage of intermediate activations or features. This creates a persistent trade-off between faithful reconstruction, editability, and interactive efficiency.

Inversion-free direct editing. Inversion-free methods avoid reconstructing an exact noise code by coupling source and target predictions through consistency sampling or conditional transport (Xu et al., 2024; Kulikov et al., 2025). This direct view removes a major computational bottleneck and exposes the editing displacement itself as the central object of study. Trajectory-aware regularization can further preserve source structure by constraining conditional dynamics (Kim et al.,

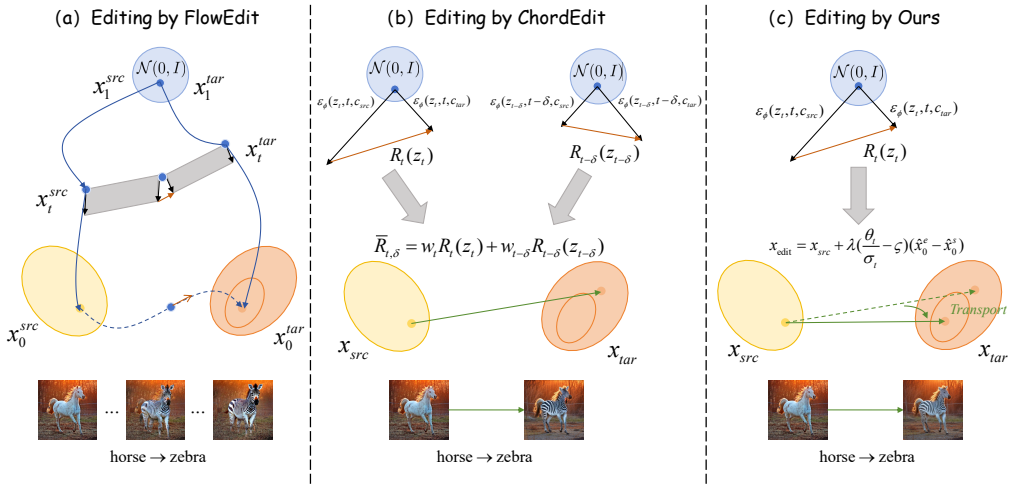


Figure 2: Editing pipelines, illustrated by a horse-to-zebra edit. (a) FlowEdit accumulates source–target field differences along a multi-step path. (b) ChordEdit combines residuals at t and $t-\delta$ into one update, using two temporal supports and four condition–time branches. (c) ECT uses only $R_t(z_t)$ from two text conditions at one shared noisy state. The same predictions determine the angular residual g_t and endpoint displacement d_t: angular transport gives $\widehat{a}_t = -\theta_t g_t$, and endpoint calibration gives $u_t^* = \widehat{a}_t - \zeta d_t$. A single update $x_{\text{edit}} = x_{\text{src}} + \lambda u_t^*$ then produces the edit (Equation 22). The sketches are schematic; the update rescales a shared edit axis.

2026), and recent rectified-flow editors refine this direct transport through noise-domain alignment, structure-aware noise rectification, step-level inversion accuracy, velocity-discrepancy modeling, or frequency-interactive attention (Xie et al., 2026; Jiang et al., 2026; Yang & Ye, 2026; Beaudouin et al., 2025; Yang et al., 2026; Xu et al., 2025), but existing formulations generally observe, integrate, or correct the state along multiple ODE locations. We ask whether one temporal observation can provide both a global edit and its reliability correction.

One-step generation and editing. Distillation, adversarial training, and rectified flow have compressed high-quality generation into one or a few steps (Liu et al., 2022; Lipman et al., 2022; Song et al., 2023; Liu et al., 2023; Sauer et al., 2024; Nguyen & Tran, 2024; Dao et al., 2024), building on large-scale latent and rectified-flow backbones (Rombach et al., 2022; Podell et al., 2023; Esser et al., 2024; Yan et al., 2024; Greenberg, 2025). Editing is harder than unconditional generation because the single large update must simultaneously introduce target semantics and preserve input-specific content. Existing efficient editors therefore retain several generation steps or stabilize a final update through residuals evaluated at multiple times (Deutch et al., 2024; Gong et al., 2025; Nguyen et al., 2025; Lu et al., 2026; Li et al., 2026). We use the VP angular horizon to obtain nominal global transport from one local residual, then calibrate it with the same query's clean endpoint signal.

3 Method

ECT denotes the complete editor, including both analytic stages. We first convert a source-target conditional residual at one noise level into a nominal endpoint transport using VP angular geometry. We then apply a proximal endpoint correction to that transport under an endpoint-exposure constraint. Figure 2 contrasts this computation with multi-step FlowEdit and two-time ChordEdit, while Figure 4 illustrates the motivation for calibrating a single-step displacement.

3.1 From Two-Time Averaging to One Evaluation

Given a source image I_{src}, source prompt c_{src}, and target prompt c_{tar}, a pretrained VAE encoder produces $x_{\text{src}} = E(I_{\text{src}})$. We seek a direct edit with one latent update,

$$x_{\text{edit}} = x_{\text{src}} + u_t. \tag{1}$$

Here u_t is the single latent edit vector, applied once at noise level t, that carries the source latent to its edited version; the whole method is a recipe for constructing it from one query. A residual observed at one time is local, and naively enlarging it into a global update can over-edit structure. A representative remedy averages residuals from two noise levels (Lu et al., 2026), but even batched into one call it still uses two temporal supports and four branches.

More generally, let the conditional editing residual at time t be

$$R_t(z_t) = \epsilon_\phi(z_t, t, c_{\text{tar}}) - \epsilon_\phi(z_t, t, c_{\text{src}}). \tag{2}$$

The residual R_t is the only condition-specific signal a single query provides, so every editor below—including ours—is ultimately a rule for turning R_t into the update u_t of Equation 1. A two-time estimator constructs a stabilized direction in the generic form

$$\bar{R}_{t,\delta} = w_t R_t(z_t) + w_{t-\delta} R_{t-\delta}(z_{t-\delta}), \qquad w_t + w_{t-\delta} = 1, \tag{3}$$

where the second observation reduces the uncertainty of extrapolating a local residual into one large update. Effective as it is, this couples editing to two schedule locations and doubles the condition-time width, even though the update is applied once.

Figure 4(a,b) illustrates the motivation: a direct single-step drift can produce dispersed endpoints, whereas combining two residuals can stabilize the final update. Panels (c,d) illustrate the complementary need to calibrate a coherent but displaced set of endpoints. These are distribution-level cartoons, not measured trajectories or literal vector identities; our shared-state construction below performs the calibration by changing the magnitude along one edit axis.

We restrict u_t to source-target predictions at one shared noisy state, formed by the VP process:

$$z_t = \alpha_t x_{\text{src}} + \sigma_t \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \qquad \alpha_t^2 + \sigma_t^2 = 1, \tag{4}$$

we evaluate

$$\hat{\epsilon}_s = \epsilon_\phi(z_t, t, c_{\text{src}}), \qquad \hat{\epsilon}_e = \epsilon_\phi(z_t, t, c_{\text{tar}}). \tag{5}$$

Using the same z_t and t makes the prediction difference a comparison of text conditions at a fixed latent state. The common z_t term cancels algebraically in the endpoint difference; independent noise draws would introduce a state difference as well. We convert this residual into a finite-horizon transport and calibrate its uncertainty through a deterministic map

$$u_t = \mathcal{T}(z_t, t, \hat{\epsilon}_s, \hat{\epsilon}_e; \alpha_t, \sigma_t) \tag{6}$$

that contains no model output from a second temporal support.

3.2 Single-Evaluation Angular Transport

To construct $\mathcal{T}$ in Equation 6 from VP geometry, define

$$\theta_t = \operatorname{atan2}(\sigma_t, \alpha_t), \qquad z_\theta = \cos\theta\, x_0 + \sin\theta\, \epsilon. \tag{7}$$

Differentiating this parameterization gives

$$\frac{dz_\theta}{d\theta} = \cos\theta\, \epsilon - \sin\theta\, x_0. \tag{8}$$

The remaining angular interval to the clean endpoint is therefore θ_t; this parameter interval need not equal the latent-space Euclidean arc length. For condition c, one epsilon prediction at z_t determines both its clean endpoint estimate and the corresponding angular tangent,

$$\hat{x}_0^c = \frac{z_t - \sigma_t \hat{\epsilon}_c}{\alpha_t}, \qquad \hat{v}_\theta^c = \alpha_t \hat{\epsilon}_c - \sigma_t \hat{x}_0^c. \tag{9}$$

The source-target angular residual is

$$g_t = \hat{v}_\theta^e - \hat{v}_\theta^s. \tag{10}$$

Here g_t is the local angular-velocity change induced by replacing the source condition with the target. Let $g(\theta)$ denote this residual along the unknown editing trajectory. The ideal action is

$$a_t^* = \int_{\theta_t}^{0} g(\theta)\, d\theta. \tag{11}$$

Although $g(\theta)$ away from θ_t is unobserved, its integration interval is fixed by the VP schedule. We therefore use the frozen-tangent approximation $g(\theta) \approx g_t$, giving the nominal transport

$$\widehat{a}_t = \int_{\theta_t}^{0} g_t\, d\theta = -\theta_t g_t. \tag{12}$$

This uses the known angular horizon without querying a second temporal support. Section 3.4 gives the rescaling $\theta_t/\sigma_t \geq 1$, and Appendix A explains its interpretation and limits.

We first bound the quadrature error for an exact local residual; model prediction error is separate.

Proposition 1 (single-evaluation nominal transport). Let $g(\theta)$ be the true source-target angular residual on $[0, \theta_t]$, assume it is L-Lipschitz, and take $g_t = g(\theta_t)$. Then

$$\|a_t^* - \widehat{a}_t\|_2 \leq \frac{L}{2}\theta_t^2. \tag{13}$$

The approximation is exact for a constant residual. A prediction error $\|g_t - g(\theta_t)\|_2 \leq \eta$ adds $\theta_t \eta$ to the bound (Appendix A); the bound does not determine its direction.

3.3 Proximal Endpoint Correction

Angular transport sets the nominal magnitude from schedule geometry by extrapolating one noisy estimate. Over a long horizon, the error in Proposition 1 can cause over- or under-shooting. We moderate the transport using the same predictions' clean endpoint displacement, requiring no extra evaluation:

$$d_t = \hat{x}_0^e - \hat{x}_0^s. \tag{14}$$

We use d_t as an endpoint-sensitive direction: $\langle d_t, u \rangle$ measures how strongly a candidate action advances along the endpoint displacement. Given an exposure budget b_t, ECT picks the action closest to the nominal transport while limiting this component,

$$u_t^* = \arg\min_u \frac{1}{2}\|u - \widehat{a}_t\|_2^2 \quad \text{s.t.} \quad \langle d_t, u \rangle \leq b_t. \tag{15}$$

This preserves the semantic direction from angular transport, changing it only as much as endpoint exposure requires. Its Lagrangian is

$$\mathcal{L}(u, \zeta) = \frac{1}{2}\|u - \widehat{a}_t\|_2^2 + \zeta\left(\langle d_t, u \rangle - b_t\right), \qquad \zeta \geq 0. \tag{16}$$

The associated KKT conditions are

$$\begin{aligned} u - \widehat{a}_t + \zeta d_t &= 0, \\ \langle d_t, u \rangle - b_t &\leq 0, \\ \zeta\big(\langle d_t, u \rangle - b_t\big) &= 0. \end{aligned} \tag{17}$$

Consequently, the unique closed-form correction is

$$u_t^* = \widehat{a}_t - \zeta d_t = -\theta_t g_t - \zeta d_t. \tag{18}$$

If the nominal transport satisfies the constraint, $\zeta = 0$ and it is unchanged; otherwise,

$$\zeta = \frac{\langle d_t, \widehat{a}_t \rangle - b_t}{\|d_t\|_2^2}, \qquad b_t = \langle d_t, \widehat{a}_t \rangle - \zeta \|d_t\|_2^2. \tag{19}$$

The implementation uses ζ directly, equivalent to the implicit budget in Equation 19. This analytic shrinkage requires no further model evaluation. Appendix C gives a conditional risk interpretation, rather than a guarantee for arbitrary model errors; Appendix B gives the projection derivation.

3.4 Closed-Form Editor and Complexity

Combining transport (Section 3.2) and correction (Section 3.3) gives a closed-form editor. The shared z_t links g_t and d_t algebraically, eliminating numerical optimization. Equation 9 yields

$$d_t = -\frac{\sigma_t}{\alpha_t}\left(\hat{\epsilon}_e - \hat{\epsilon}_s\right), \qquad g_t = \frac{1}{\alpha_t}\left(\hat{\epsilon}_e - \hat{\epsilon}_s\right). \tag{20}$$

Therefore,

$$g_t = -\frac{1}{\sigma_t} d_t. \tag{21}$$

Substituting Equation 21 into Equation 18 yields the final editor,

$$\boxed{x_{\text{edit}} = x_{\text{src}} + \lambda\left(\frac{\theta_t}{\sigma_t} - \zeta\right)\left(\hat{x}_0^e - \hat{x}_0^s\right)}. \tag{22}$$

Here λ is the user-facing edit-strength coefficient, θ_t/σ_t the schedule-fixed transport gain, and ζ the endpoint-exposure shrinkage. The complete procedure is therefore

$$(x_{\text{src}}, c_{\text{src}}, c_{\text{tar}}) \xrightarrow{\text{one } t} (\hat{\epsilon}_s, \hat{\epsilon}_e) \xrightarrow{\text{closed form}} x_{\text{edit}}, \tag{23}$$

with no numerical solver between model prediction and latent update. Adding the endpoint difference to x_{src} also retains the source reconstruction residual $x_{\text{src}} - \hat{x}_0^s$; replacing the source directly with $\hat{x}_0^e$ would discard this input-specific component.

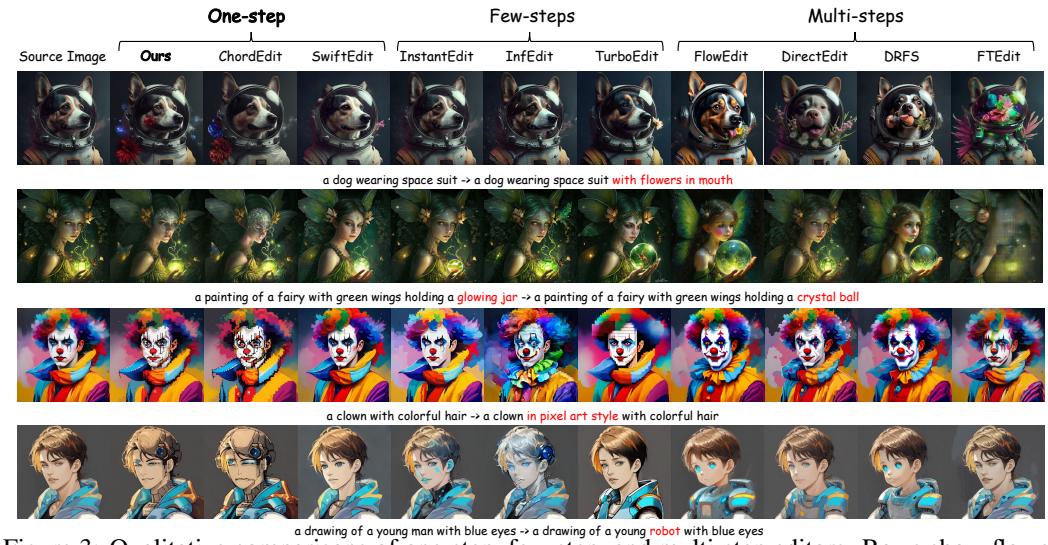


Figure 3: Qualitative comparisons of one-step, few-step, and multi-step editors. Rows show flower addition, crystal-ball replacement, pixel-art stylization, and robot transformation. Source images and instructions illustrate the preservation–alignment trade-offs.

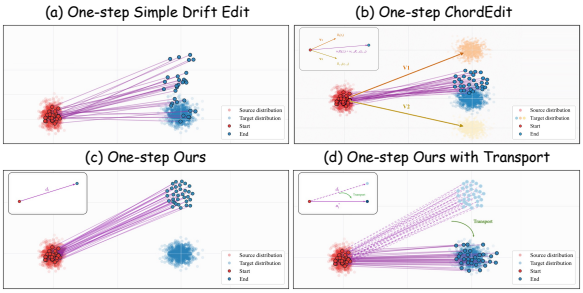


Figure 4: Schematic motivation for single-step calibration. (a) Simple drift produces dispersed endpoints. (b) ChordEdit combines two residuals. (c) Coherent updates can still miss the target distribution. (d) Transport and correction calibrate the endpoints. Direction changes abstract high-dimensional behavior; the exact ECT update is collinear with d_t (Equation 21).

ECT changes the magnitude along one edit axis. The nominal gain $\theta_t/\sigma_t \geq 1$ follows from the frozen-tangent approximation, and ζ moderates that gain; neither operation supplies a new direction. With $\hat{x}_0^s \approx x_{\text{src}}$, the update is approximately an interpolation toward the target endpoint when $0 \leq \lambda(\theta_t/\sigma_t - \zeta) \leq 1$. We evaluate this trade-off empirically.

The method constructs z_t at one noise level, evaluates the two text conditions in one batch, and applies Equation 22 once, with neither inversion nor ODE integration. Section 4.3 reports temporal supports, condition-time branches, batched calls, latent updates, and measured efficiency.

4 Experiments

4.1 Experimental Setup

Benchmarks and backbones. We evaluate on PIE-Bench and SNR-Bench with SD-Turbo (SD-T) (Sauer et al., 2024) and SwiftBrush-v2 (SB-v2) (Dao et al., 2024). The main setting is strictly single-evaluation—one temporal support, two text-conditioned branches in one denoiser batch, one latent update—with $t = 0.75$, $\zeta = 0.7$, and $\lambda = 1$ unless stated otherwise. Appendix J reports seed and parameter studies. Extra-NFE variants appear only in the ablations of Table 3(c) and Appendix J.2, not in the main results.

Metrics. We report MSE, PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018), all computed after masking the edit region so they measure preservation of the unedited content, and CLIP cosine similarity (Radford et al., 2021) for target-prompt alignment on the full edited image. ImgEdit-Judge (Ye et al., 2026) additionally scores Content and Identity Preservation (CP), Visual Quality and Artifact Control (VQ), and Physical and Perceptual Coherence (PC) on independent 1–5 scales. Appendices D and F give the preprocessing, formulae, aggregation, and full judge protocol. System efficiency uses synchronized per-image latency, time-weighted CPU-process utilization and resident set size (RSS), and peak PyTorch-reserved VRAM, under identical backbone, resolution,

process scope, and instrumentation per comparison. These are resource demands, not quality scores: lower is lighter. Appendix E defines every column and records temporal supports, condition-time branches, batched denoiser calls, and latent updates.

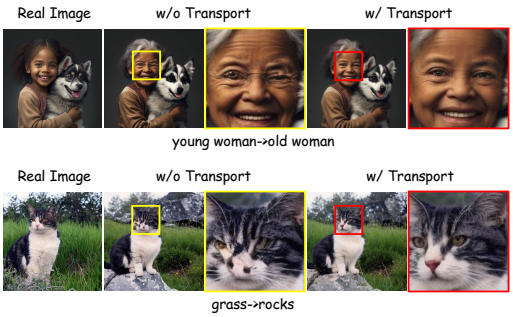


Figure 5: Qualitative effects of angular transport on two representative cases. From left to right: source, result without transport and its enlarged crop, then the "w/ Transport" result and its enlarged crop.

Baselines. Our central comparison is the two-temporal-support ChordEdit estimator (Lu et al., 2026) on the same one-step backbone. The main tables retain all non-ablation algorithm–model combinations from both benchmarks: multi-step DirectEdit (Yang & Ye, 2026), DNAEdit (Xie et al., 2026), SNR-Edit (Jiang et al., 2026), FlowEdit (Kulikov et al., 2025), DRFS (Beaudouin et al., 2025), FIA-Edit (Yang et al., 2026), and FTEdit (Xu et al., 2025); few-step (4-step) TurboEdit (Deutch et al., 2024), InfEdit (Xu et al., 2024), and InstantEdit (Gong et al., 2025); and one-step SwiftEdit (Nguyen et al., 2025) and ChordEdit—across the FLUX (Greenberg, 2025), SD3.5 (Esser et al., 2024), SDXL-Turbo/SD-Turbo (Sauer et al., 2024), SD1.4 (Rombach et al., 2022), PeRFlow (Yan et al., 2024), and SwiftBrush-v2 (Dao et al., 2024) backbones listed in each Model column. The transport ablations are omitted here; matched ChordEdit/ECT pairs are highlighted to isolate temporal support and endpoint calibration.

4.2 Main Quantitative Results

Tables 1 and 2 include all reported baseline/model combinations. For display, LPIPS/MSE are scaled by 1000 and SSIM/CLIP by 100, without affecting ranks. Quality metrics average finite per-sample scores, excluding invalid or unparsed values separately for each metric (Appendices D and F). Time is in seconds per image (three decimals), CPU in integer percentages, and VRAM/Mem. in MiB; percentage changes use unrounded measurements. Rules separate multi-step, four-step, and one-step settings; one-step rows group matched ChordEdit/ECT pairs by backbone. Avg. rank summarizes all 12 metrics within each table using displayed values, indicated directions, and average ranks for ties; it is not a quality metric. Both methods use one batched call and one latent update, with two temporal supports for ChordEdit and one for ECT.

ECT improves PSNR and all three judged scores on both PIE-Bench backbones. SD-Turbo also improves MSE, LPIPS, and SSIM, while SwiftBrush-v2 shows a small preservation trade-off. On SNR-Bench, all four preservation metrics and all judged scores improve on both backbones, with modest CLIP decreases. These paired results support improved source fidelity and judged quality with small alignment trade-offs, although preservation gains depend on the benchmark. Figure 3 shows qualitative comparisons.

The SB-v2 PSNR/MSE results are compatible because PSNR applies a logarithm before averaging: higher mean PSNR need not imply lower mean MSE. Reporting both exposes differences that a single preservation score would conceal. The matched pairs also hold backbone capacity fixed, so their differences more directly reflect the editing procedure than comparisons across generators with different architectures and training objectives.

4.3 System Efficiency

Compared with matched ChordEdit rows in Tables 1 and 2, ECT reduces synchronized latency by 15.7–20.0%, time-weighted CPU-process utilization by 42.1–48.0%, and time-weighted process RSS by 17.0–21.4%. Peak reserved VRAM changes by only 0.24% (7,480 versus 7,498 MiB). These paired measurements use common instrumentation and are not comparable across hardware or process scopes (Appendix E). The reductions are consistent with halving condition-time branches, reducing input preparation, transfers, and intermediate activations. Temporal supports and branch counts therefore complement batched-call and latent-update counts when reporting efficiency.

VAE encoding/decoding and other shared work remain after reducing the branches. Two instead of four conditional branches therefore need not halve the total latency. Peak reserved VRAM includes

Table 1: PIE-Bench results. ECT has the best average rank on both one-step backbones. Blue highlights ECT, bold marks gains over matched ChordEdit, and orange marks resource savings.

Stage	Method	Model	PSNR ↑	MSE ↓	LPIPS ↓	SSIM ↑	CLIP ↑	Time ↓	VRAM ↓	CPU ↓	Mem. ↓	CP ↑	VQ ↑	PC ↑	Avg. rank ↓
Multi-step	DirectEdit	FLUX	20.00	12.89	309.71	72.24	26.14	14.690	37284	1025	6474.8	4.04	4.39	4.03	13.67
Multi-step	DNAEdit	FLUX	21.85	8.34	219.16	78.41	24.78	7.860	35881	1135	6468.6	4.02	4.45	4.11	11.13
Multi-step	SNR-Edit	FLUX	21.08	9.75	204.70	77.66	25.19	9.410	6446	2811	1333.3	4.13	4.59	4.32	8.50
Multi-step	FlowEdit	FLUX	18.68	16.63	276.07	73.25	25.81	8.280	37714	1149	6478.9	4.02	4.43	4.15	14.38
Multi-step	FlowEdit	SD3.5	19.17	14.51	267.66	73.38	27.71	3.990	21873	1048	4737.0	3.97	4.29	3.88	14.17
Multi-step	DirectEdit	SD3.5	22.85	6.12	185.50	81.10	26.13	5.400	21412	2536	4751.4	4.05	4.39	4.01	9.58
Multi-step	DNAEdit	SD3.5	22.93	6.20	199.40	80.82	25.61	3.240	19924	1114	4721.7	4.02	4.39	4.02	9.17
Multi-step	DRFS	SD3.5	20.55	10.79	223.68	76.66	27.70	5.420	19922	1234	4710.4	4.03	4.38	4.00	12.17
Multi-step	FIA-Edit	SD3.5	23.26	5.87	159.61	81.02	31.07	5.330	19930	1322	4711.4	4.05	4.43	4.09	7.08
Multi-step	FTEdit	SD3.5	20.90	10.67	275.94	79.41	23.46	31.240	26890	1380	2203.7	3.63	4.06	3.67	15.17
Multi-step	SNR-Edit	SD3.5	19.53	13.16	250.62	74.12	26.57	5.250	6448	3391	1388.5	4.07	4.38	4.02	11.54
Few-step (4-step)	TurboEdit	SDXL-T	21.44	9.49	108.60	75.08	24.66	0.610	14022	679	819.2	3.96	4.33	3.93	10.25
Few-step (4-step)	InfEdit	SD1.4	19.74	15.83	224.83	68.67	25.13	0.481	7784	419	3364.9	3.85	4.27	3.83	13.83
Few-step (4-step)	InstantEdit	PeRFlow	22.30	8.92	150.54	76.24	25.77	0.670	9156	1294	843.8	4.03	4.39	4.03	8.92
One-step	SwiftEdit	SB-v2	20.65	9.91	200.91	69.97	30.01	0.451	16224	182	4282.4	4.07	4.53	4.22	7.79
One-step	ChordEdit	SB-v2	22.04	7.13	120.49	77.27	29.70	0.265	7498	357	1298.7	3.91	4.37	4.05	7.38
One-step	ECT	SB-v2	**22.86**	7.56	122.81	77.05	**29.70**	**0.213** (-19.6%)	**7480**	**190** (-46.7%)	**1075.9** (-17.2%)	**4.03**	**4.45**	**4.13**	**4.92**
One-step	ChordEdit	SD-T	22.63	7.99	122.64	76.71	30.60	0.265	7498	357	1297.1	3.95	4.37	4.00	7.71
One-step	ECT	SD-T	**23.74**	**6.22**	**109.04**	**78.10**	30.20	**0.212** (-20.0%)	**7480**	**190** (-46.7%)	**1076.0** (-17.0%)	**4.09**	**4.65**	**4.31**	**2.67**

Table 2: SNR-Bench results. ECT has the best average rank on both one-step backbones. Blue highlights ECT, bold marks gains over matched ChordEdit, and orange marks resource savings.

Stage	Method	Model	PSNR ↑	MSE ↓	LPIPS ↓	SSIM ↑	CLIP ↑	Time ↓	VRAM ↓	CPU ↓	Mem. ↓	CP ↑	VQ ↑	PC ↑	Avg. rank ↓
Multi-step	DirectEdit	FLUX	15.96	42.66	450.31	51.22	26.30	15.101	37284	103	6676.9	4.31	4.53	4.28	12.08
Multi-step	DNAEdit	FLUX	19.52	15.05	250.75	64.82	26.22	8.698	35881	104	6691.8	4.48	4.60	4.49	6.79
Multi-step	SNR-Edit	FLUX	18.62	17.09	249.63	60.58	26.45	9.454	6446	121	8267.7	4.46	4.58	4.34	7.67
Multi-step	FlowEdit	FLUX	16.77	23.96	318.52	58.79	26.96	8.653	37714	102	7246.5	4.41	4.56	4.38	9.21
Multi-step	FlowEdit	SD3.5	19.26	14.82	237.77	64.77	28.39	3.112	21873	104	5243.0	4.45	4.59	4.44	5.46
Multi-step	DirectEdit	SD3.5	17.38	37.76	390.36	54.99	25.78	5.147	21412	108	5144.7	4.28	4.45	4.23	11.33
Multi-step	DRFS	SD3.5	20.05	13.03	213.40	66.27	27.85	5.360	19922	105	5270.1	4.50	4.69	4.48	4.75
Multi-step	FIA-Edit	SD3.5	17.86	26.76	332.32	56.92	27.05	5.371	19930	113	5076.1	4.36	4.51	4.24	9.46
Multi-step	SNR-Edit	SD3.5	18.80	15.58	239.69	62.89	28.01	5.350	6448	127	5097.5	4.34	4.56	4.43	6.13
Few-step (4-step)	TurboEdit	SDXL-T	14.77	40.80	430.38	46.42	27.62	0.745	14022	141	1651.3	4.15	4.31	4.08	12.42
Few-step (4-step)	InfEdit	SD1.4	20.87	9.97	193.72	64.71	23.37	0.434	7784	265	2777.9	4.23	4.50	4.28	7.42
Few-step (4-step)	InstantEdit	PeRFlow	14.93	41.31	423.58	47.51	27.92	0.815	9156	275	1804.3	4.10	4.36	4.05	12.75
One-step	SwiftEdit	SB-v2	16.74	30.29	348.18	49.60	24.87	0.472	16224	185	4756.5	4.26	4.44	4.28	11.71
One-step	ChordEdit	SB-v2	18.64	15.84	341.59	52.38	24.88	0.305	7498	348	1312.5	4.21	4.41	4.21	10.42
One-step	ECT	SB-v2	**19.61**	**12.59**	**314.76**	**54.48**	24.31	**0.254** (-16.7%)	**7480**	**181** (-48.0%)	**1037.1** (-21.0%)	**4.30**	**4.44**	**4.30**	**7.25**
One-step	ChordEdit	SD-T	17.86	19.32	348.59	50.57	25.76	0.305	7498	332	1316.1	4.14	4.46	4.15	10.88
One-step	ECT	SD-T	**19.02**	**14.60**	**311.84**	**53.29**	25.32	**0.257** (-15.7%)	**7480**	**192** (-42.1%)	**1035.0** (-21.4%)	**4.30**	**4.48**	**4.30**	**7.29**

cached device allocations, whereas process RSS measures host memory; their different scopes help explain why memory savings need not move together.

4.4 User Study

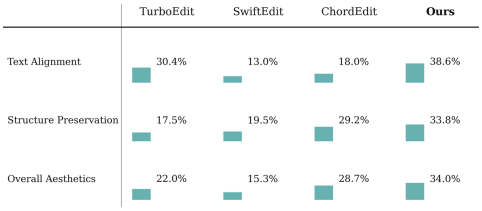


Figure 6: User-study first-place shares (%), based on mean participant scores per case, with no top ties. ECT leads in text alignment (38.6%), structure preservation (33.8%), and aesthetics (34.0%).

We compare TurboEdit, SwiftEdit, ChordEdit, and ECT in a blinded PIE-Bench study. Participants independently score anonymous outputs on assigned subsets of varying sizes from 1 to 5 for text alignment, structure preservation, and aesthetics. Methods are ranked by mean participant scores per case and criterion (Figure 6). ECT leads with first-place shares of 38.6%, 33.8%, and 34.0%, exceeding runners-up—TurboEdit for alignment and ChordEdit for structure and aesthetics—by 8.2, 4.6, and 5.3 percentage points, respectively. These descriptive differences alone do not establish statistical significance. Appendix G details the protocol.

4.5 Ablation Studies

Table 3 examines one- versus two-step ECT in Panel (a), the angular transport stage in Panel (b), and an extra neural function evaluation (NFE) in Panel (c), reporting relevant metrics. Figure 5 separately illustrates angular transport, while Appendix J.2 studies supplementary refinement noise levels.

Table 3, Panel (a), makes the temporal-support comparison explicit. Relative to two-step ECT, the one-step editor reduces latency by 17.8%/18.1%, CPU utilization by 57.4%/57.9%, and process

Table 3: Ablations on PIE-Bench: step count (a), angular transport (b), and extra NFE (c). Blue marks ECT variants, bold better values, and orange resource reductions.

(a) One- vs. two-step ECT

Model	Setting	Time (s) ↓	CPU (%) ↓	Mem. (MiB) ↓	CP ↑	VQ ↑	PC ↑
SB-v2	ChordEdit	0.265	357	1298.74	3.91	4.37	4.05
SB-v2	ECT (2-step)	0.259	446	1382.99	3.99	4.42	4.10
SB-v2	ECT (1-step)	**0.213** (-17.8%)	**190** (-57.4%)	**1075.93** (-22.2%)	**4.03**	**4.45**	**4.13**
SD-T	ChordEdit	0.265	357	1297.07	3.95	4.37	4.00
SD-T	ECT (2-step)	0.259	452	1386.67	4.04	4.44	4.07
SD-T	ECT (1-step)	**0.212** (-18.1%)	**190** (-57.9%)	**1075.96** (-22.4%)	**4.09**	**4.65**	**4.31**

(b) Transport stage

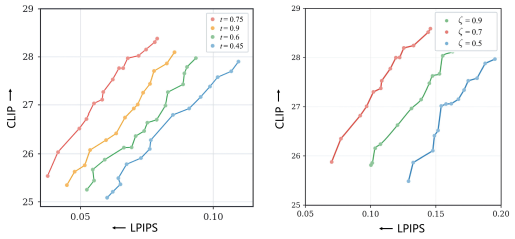
Model	Variant	PSNR ↑	LPIPS ↓	CLIP ↑	VQ ↑	PC ↑
SB-v2	w/o transport	22.7	124.2	**29.8**	4.4	4.0
SB-v2	ECT (full)	**22.9**	**122.8**	29.7	**4.5**	**4.1**
SD-T	w/o transport	22.4	127.1	**30.7**	4.4	4.0
SD-T	ECT (full)	**23.7**	**109.0**	30.2	**4.7**	**4.3**

(c) Extra NFE

Model	Variant	PSNR ↑	LPIPS ↓	CLIP ↑	VQ ↑	PC ↑
SB-v2	w/o extra NFE	**22.9**	142.0	22.8	**4.5**	**4.2**
SB-v2	w/ extra NFE	**22.9**	**122.8**	**29.7**	**4.5**	4.1
SD-T	w/o extra NFE	**26.0**	**64.1**	28.6	4.3	3.9
SD-T	w/ extra NFE	23.7	109.0	**30.2**	**4.7**	**4.3**

RSS by 22.2%/22.4% on SB-v2/SD-T, while improving CP, VQ, and PC for both backbones. This establishes the one-step setting as the more favorable quality–efficiency operating point.

Table 3(b) shows that including angular transport improves PSNR and LPIPS on both backbones, while both reported image-edit ratings increase. At the displayed precision, CLIP decreases from 29.8 to 29.7 on SB-v2 and from 30.7 to 30.2 on SD-T, so the preservation improvement comes with a modest alignment trade-off. Adding the extra NFE in Panel (c) raises SB-v2's CLIP from 22.8 to 29.7 and lowers LPIPS from 142.0 to 122.8, while VQ remains 4.5 and PC decreases from 4.2 to 4.1. On SD-T, CLIP, VQ, and PC improve, while PSNR falls from 26.0 to 23.7 and LPIPS increases from 64.1 to 109.0. This supplementary extra-NFE comparison shows a trade-off between background fidelity and semantic editing on SD-T.



(a) Noise level t (b) Correction ζ

Figure 7: Preservation–alignment trade-offs on PIE-Bench. Lower LPIPS and higher CLIP are preferable; LPIPS is unscaled. (a) $t = 0.75$ gives lower LPIPS at comparable CLIP than the other plotted evaluation times. (b) $\zeta = 0.7$ gives the most favorable plotted trade-off.

Parameter sensitivity. Figure 7(a) compares the LPIPS–CLIP trade-offs at four evaluation noise levels. Over their shared CLIP range, $t = 0.75$ achieves lower LPIPS than $t = 0.45$, 0.6, and 0.9, supporting its use as the default. Panel (b) similarly favors the intermediate correction coefficient $\zeta = 0.7$ over 0.5 and 0.9. These plots measure the balance between preservation and alignment rather than PSNR or a single metric optimum. Appendix J reports the seed and remaining parameter controls. In particular, Table 10 shows that $\lambda = 1$ has the best preservation in the tested sweep, while $\lambda = 2$ gains 0.50 CLIP points at a cost of 2.690 dB PSNR and 37.636 scaled LPIPS units. Thus, $\lambda = 1$ favors preservation over maximizing CLIP alone.

The seed control tests dependence on a favorable noisy state. Across eleven seeds, the worst result for each preservation metric still outperforms matched ChordEdit (Appendix J.1), supporting stable benchmark averages without implying identical edits across seeds.

5 Conclusion

We presented Endpoint-Calibrated Transport (ECT), a training-free approach to inversion-free image editing that uses one shared noisy state, one batched denoiser evaluation, and one latent update. The method turns the source–target prediction difference into a schedule-aware angular transport and uses the clean endpoint displacement from the same query to calibrate that transport, so semantic motion and structural control are obtained without inversion, a second temporal support, ODE integration, or iterative optimization. Across the supplied benchmark summaries and matched backbones, ECT improves PSNR in every one-step comparison, improves all reported preservation metrics on SNR-Bench, raises the judged image-edit scores throughout, and exposes the PIE-Bench SwiftBrush-v2 preservation trade-off rather than hiding it. At the same time, the matched tables show substantial reductions in synchronized latency, CPU-process utilization, and process RSS. These properties make single-temporal-support editing a compact and practical operating point while keeping the method's assumptions and quality–efficiency trade-off explicit.

Limitations and future work. The current derivation assumes epsilon-prediction VP one-step backbones and a shared noisy state, and therefore does not yet cover non-VP schedules, velocity-prediction models, or spatially adaptive constraints. The reported efficiency numbers also depend on hardware, process scope, and measurement protocol. Future work will extend the formulation to these settings and evaluate broader controls once the corresponding runs are available.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, manuscript organization, and consistency checking. The authors reviewed all derivations, implementation correspondences, experimental values, citations, and claims, and take responsibility for the content.

ETHICS STATEMENT

This work includes an anonymous perceptual study of generated images. The rating interface requests no names, email addresses, or demographic attributes; the study database stores a pseudonymous browser identifier, ratings, and UTC timestamps. Like other generative editing systems, the method may be misused to create misleading visual content. Deployment should include provenance, consent, and disclosure practices appropriate to the application.

REPRODUCIBILITY STATEMENT

Section 3 specifies the complete closed-form method and assumptions. Appendix A proves the single-evaluation approximation bound and Appendix B derives the endpoint projection. Appendices D–G define automatic image metrics, system-resource accounting, ImgEdit-Judge evaluation, and the user-study protocol. Appendix J reports the seed and λ controls together with the exact values used to produce every table. The supplementary material will include configuration files, benchmark preprocessing, per-sample outputs, and commands for both backbones.

REFERENCES

Gaspard Beaudouin, Minghan Li, Jaeyeon Kim, Sunghoon Yoon, and Mengyu Wang. Delta velocity rectified flow for text-to-image editing. *arXiv e-prints*, pp. arXiv–2509, 2025.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions, 2023. URL https://arxiv.org/abs/2211.09800.

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 *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22503–22513. IEEE, 2023.

Trung Dao, Thuan Hoang Nguyen, Thanh Le, Duc Vu, Khoi Nguyen, Cuong Pham, and Anh Tran. Swiftbrush v2: Make your one-step diffusion model better than its teacher. In *European Conference on Computer Vision*, pp. 176–192. Springer, 2024.

Gilad Deutch, Rinon Gal, Daniel Garibi, Or Patashnik, and Daniel Cohen-Or. Turboedit: Text-based image editing using few-step diffusion models. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–12, 2024.

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 *Forty-first international conference on machine learning*, 2024.

Yiming Gong, Zhen Zhu, and Minjia Zhang. Instantedit: Text-guided few-step image editing with piecewise rectified flow. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16808–16817. IEEE, 2025.

Or Greenberg. Demystifying flux architecture. *arXiv preprint arXiv:2507.09595*, 2025.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

Lifan Jiang, Boxi Wu, Yuhang Pei, Tianrun Wu, Yongyuan Chen, Yan Zhao, Shiyu Yu, and Deng Cai. Snr-edit: Structure-aware noise rectification for inversion-free flow-based editing. *arXiv preprint arXiv:2601.19180*, 2026.

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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19721–19730. IEEE, 2025.

Minghan Li, Jeremy Moebel, and Mengyu Wang. Rethinking one-step image editing through chordedit: Reproduction, simplification, and new insights. *arXiv preprint arXiv:2606.14042*, 2026.

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

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. URL https://arxiv.org/abs/2209.03003.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The twelfth international conference on learning representations*, 2023.

Liangsi Lu, Xuhang Chen, Minzhe Guo, Shichu Li, Jingchao Wang, and Yang Shi. Chordedit: One-step low-energy transport for image editing. *arXiv preprint arXiv:2602.19083*, 2026.

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 (2022). *arXiv preprint arXiv:2108.01073*, 2022.

Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2063–2072. IEEE, 2025.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6038–6047. IEEE, 2023.

Thuan Hoang Nguyen and Anh Tran. Swiftbrush: One-step text-to-image diffusion model with variational score distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7807–7816. IEEE, 2024.

Trong-Tung Nguyen, Quang Nguyen, Khoi Nguyen, Anh Tran, and Cuong Pham. Swiftedit: Lightning fast text-guided image editing via one-step diffusion. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21492–21501. IEEE, 2025.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023. URL https://arxiv.org/abs/2307.01952.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pp. 87–103. Springer, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1921–1930. IEEE, 2023.

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.

Chenxi Xie, Minghan Li, Shuai Li, Yuhui Wu, Qiaosi Yi, and Lei Zhang. Dnaedit: Direct noise alignment for text-guided rectified flow editing. *Advances in Neural Information Processing Systems*, 38:124456–124474, 2026.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28479–28489. IEEE, 2025.

Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with language-guided diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9454–9461. IEEE, 2024.

Hanshu Yan, Xingchao Liu, Jiachun Pan, Jun Hao Liew, Qiang Liu, and Jiashi Feng. Perflow: Piecewise rectified flow as universal plug-and-play accelerator. *Advances in Neural Information Processing Systems*, 37:78630–78652, 2024.

Desong Yang and Mang Ye. Directedit: Step-level accurate inversion for flow-based image editing. *arXiv preprint arXiv:2605.02417*, 2026.

Kaixiang Yang, Boyang Shen, Xin Li, Yuchen Dai, Yuxuan Luo, Yueran Ma, Wei Fang, Qiang Li, and Zhiwei Wang. Fia-edit: Frequency-interactive attention for efficient and high-fidelity inversion-free text-guided image editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11613–11621, 2026. No. 14.

Yang Ye, Xianyi He, Zongjian Li, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, Li Yuan, et al. Imgedit: A unified image editing dataset and benchmark. *Advances in Neural Information Processing Systems*, 38, 2026.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

A Angular Derivations and Approximation Error

Differentiating the VP angular path in Equation 7 gives

$$\frac{dz_\theta}{d\theta} = \cos\theta\,\epsilon - \sin\theta\,x_0, \tag{24}$$

which yields Equation 9 after substituting the model estimates. For the quadrature bound, write

$$a_t^* = \int_{\theta_t}^{0} g(\theta)d\theta, \qquad \widehat{a}_t = -\theta_t g(\theta_t). \tag{25}$$

If $\|g(\theta_1) - g(\theta_2)\|_2 \le L|\theta_1 - \theta_2|$, then

$$\|a_t^* - \widehat{a}_t\|_2 \le \int_0^{\theta_t} \|g(\theta) - g(\theta_t)\|_2 d\theta \tag{26}$$

$$\le \int_0^{\theta_t} L(\theta_t - \theta) d\theta = \frac{L}{2}\theta_t^2. \tag{27}$$

Angular-horizon interpretation. The nominal transport rescales the naive one-step edit by a factor fixed entirely by the schedule. From the shared state (Equation 20), the single-step endpoint displacement has magnitude $\propto \sigma_t/\alpha_t$ while the transport has magnitude $\propto \theta_t/\alpha_t$, so

$$\|\widehat{a}_t\|_2 = \frac{\theta_t}{\sigma_t}\|d_t\|_2 = \frac{\theta_t}{\sin\theta_t}\|d_t\|_2 \ge \|d_t\|_2. \tag{28}$$

Under the frozen-tangent approximation, this factor replaces $\sin\theta_t$ with θ_t and tends to one as $\theta_t \to 0$. It does not establish that d_t underestimates the true action or that rescaling always reduces its error. The interval spans an angle, not generally the latent trajectory's Euclidean arc length.

Prediction error and shrinkage. For a model residual g_t with $\|g_t - g(\theta_t)\|_2 \le \eta$, the triangle inequality gives $\|a_t^* + \theta_t g_t\|_2 \le L\theta_t^2/2 + \theta_t\eta$. Thus the quadrature bound alone does not control model error. The shrinkage analysis below assumes unbiased noise.

B Endpoint Projection and KKT Conditions

The Lagrangian of Equation 15 is

$$\mathcal{L}(u, \zeta) = \frac{1}{2}\|u - \widehat{a}_t\|_2^2 + \zeta(\langle d_t, u\rangle - b_t), \qquad \zeta \ge 0. \tag{29}$$

Stationarity gives $u_t^* = \widehat{a}_t - \zeta d_t$; for active constraints ($d_t \neq 0$), complementary slackness gives

$$\zeta_t^* = \frac{\langle d_t, \widehat{a}_t\rangle - b_t}{\|d_t\|_2^2}, \qquad b_t = \langle d_t, \widehat{a}_t\rangle - \zeta_t^*\|d_t\|_2^2. \tag{30}$$

For the degenerate case $d_t = 0$, we bypass the quotient and set $\zeta = 0$. Thus a fixed ζ is equivalent to a sample-dependent implicit exposure budget. Conversely, a specified budget yields an adaptive dual strength from the same observation.

C Shrinkage-Risk Interpretation

For the final editor in Equation 22, write $u = \kappa d_t$ with $\kappa = \lambda(\theta_t/\sigma_t - \zeta)$. Assume $d_t = d^* + e$, with $\mathbb{E}[e] = 0$, $\mathbb{E}\|e\|_2^2 = V$, and $\|d^*\|_2^2 = S$. Then u has risk

$$\mathbb{E}\|u - d^*\|_2^2 = (1-\kappa)^2 S + \kappa^2 V. \tag{31}$$

For $0 < \kappa < 1$, this is lower than the uncorrected risk V whenever

$$\frac{V}{S} > \frac{1-\kappa}{1+\kappa}. \tag{32}$$

This result compares the final coefficient κd_t with the unscaled estimator d_t for the target d^*. It motivates conservative endpoint calibration under the stated assumptions; it does not identify d^* with the trajectory integral a_t^* or establish an unconditional improvement over the nominal transport $(\theta_t/\sigma_t)d_t$.

D Automatic Image Metrics

Common preprocessing and scope. Let $S, T \in [0,1]^{3\times H\times W}$ denote the source and edited RGB images. If the prediction is larger than the source, we use the benchmark-compatible bottom-right crop; an incompatible size is excluded. Let $M \in \{0,1\}^{H\times W}$ be the edit mask ($M = 1$ in the intended edit region). Following the evaluator, all four image boundaries are also set to one before defining $B = 1 - M$. The preservation inputs are $S_B = S \odot B$ and $T_B = T \odot B$, with B broadcast over channels. MSE, PSNR, SSIM, and LPIPS measure *background preservation*; CLIP measures *target alignment* on the full edited image. Samples with no evaluable background are invalid.

Definitions. The evaluator computes

$$\mathrm{MSE}(S_B, T_B) = \frac{1}{3HW} \sum_{c,h,w} (S_{B,c,h,w} - T_{B,c,h,w})^2, \tag{33}$$

$$\mathrm{PSNR}(S_B, T_B) = 10 \log_{10} \frac{1}{\mathrm{MSE}(S_B, T_B)}. \tag{34}$$

The denominator is the full $3HW$, not the number of background elements: the edit region is zero in both inputs and contributes no error but remains in the normalization. PSNR is in dB because the data range is one; an exact match has MSE 0 and theoretical PSNR $+\infty$.

For corresponding local windows x and y, SSIM is (Wang et al., 2004)

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{35}$$

with $C_j = (K_j L)^2$ and $L = 1$; the implementation averages over windows and channels. Windows crossing the mask boundary can therefore reflect boundary structure. For LPIPS, define $\widetilde{S}_B = 2S_B - 1$ and $\widetilde{T}_B = 2T_B - 1$. In its standard notation (Zhang et al., 2018),

$$\mathrm{LPIPS}(\widetilde{S}_B, \widetilde{T}_B) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \left\| w_l \odot \left(\hat{\phi}_l(\widetilde{S}_B)_{h,w} - \hat{\phi}_l(\widetilde{T}_B)_{h,w} \right) \right\|_2^2, \tag{36}$$

where $\hat{\phi}_l$ is the channel-normalized feature at layer l and w_l is its learned channel weight; our evaluator uses the SqueezeNet backbone and released learned weights. CLIP uses cosine similarity:

$$\mathrm{CLIP}(T, e) = \frac{v_I(T)^\top v_T(e)}{\|v_I(T)\|_2 \|v_T(e)\|_2}, \tag{37}$$

Here the embeddings represent the full edited image and target instruction e, computed with `openai/clip-vit-base-patch32` (Radford et al., 2021). We select e from the first non-empty field in this order: `editing_prompt`, `edited_prompt`, `target_prompt`, then `editing_instruction` if the preceding fields are empty.

Table 4: Automatic image metrics and their interpretation.

Metric	Direction	Input	Primary interpretation
MSE	↓	Masked source/output	Mean pixel error
PSNR	↑	Masked source/output	Log pixel fidelity
SSIM	↑	Masked source/output	Local structure preservation
LPIPS	↓	Masked source/output	Learned perceptual distance
CLIP	↑	Full output/instruction	Target semantic alignment

Aggregation. Each metric is computed per sample and then averaged over its finite values, both overall and per edit category. NaN or failed values are excluded rather than replaced by zero; if a metric has no valid samples, its aggregate is NaN. We do not combine preservation and alignment into a single score. For display, tables scale MSE and LPIPS by 1000, and SSIM and CLIP by 100.

E Inference-Time and Resource Accounting

All reported efficiency measurements use an NVIDIA H20 GPU.

Measurement window. For sample i, the high-resolution timer surrounds the full editing pipeline call. A CUDA synchronization is issued before recording t_i^{end}, so $d_i = t_i^{\mathrm{end}} - t_i^{\mathrm{start}}$ includes completed GPU work. The window excludes source-image loading, tensor-to-image conversion, output writing, logging, and metric evaluation. Skipped or failed calls are omitted. Mean latency is

$$\bar{d} = \frac{1}{N} \sum_{i=1}^{N} d_i \quad \text{seconds per image.} \tag{38}$$

CPU and host memory. The process is sampled at approximately 0.1-s intervals during the measured inference windows. After priming `psutil.Process.cpu_percent`, the reported CPU value is the time-weighted process utilization

$$\bar{p} = \frac{\sum_k p_k \Delta t_k}{\sum_k \Delta t_k}. \tag{39}$$

The process convention assigns 100% to one fully occupied logical core; hence a multithreaded process can exceed 100%. This is not the 0–100% system-wide CPU percentage. Mem. is the process resident-set size (RSS), approximated as a left-endpoint time average,

$$\bar{m} = \frac{\sum_k m_{k-1} \Delta t_k}{\sum_k \Delta t_k}, \qquad \mathrm{MiB}(\bar{m}) = \bar{m}/2^{20}. \tag{40}$$

This is process-resident memory, not system-wide memory or a percentage of installed RAM.

GPU memory. After model loading, the evaluator records PyTorch's allocated and reserved baselines. Immediately before the first measured inference it resets CUDA peak statistics; after all measured samples it reads `max_memory_allocated` and `max_memory_reserved`. VRAM in the main tables is the latter divided by 2^{20}, i.e., the peak memory held by the PyTorch caching allocator, including cached blocks. It is therefore generally no smaller than live-tensor allocated memory and is neither mean VRAM occupancy nor GPU compute utilization. We also record the allocated peak and its increase over the model-loaded baseline; VRAM reports only the reserved peak.

Table 5: Resource metrics reported in the main tables.

Column	Statistic	Scope	Unit
Time	Arithmetic mean of d_i	Pipeline call	s/image
CPU	Time-weighted process utilization	Inference windows	core-%
Mem.	Time-weighted process RSS	Inference windows	MiB
VRAM	Peak PyTorch reserved memory	Measured run	MiB

Computational structure and comparability. We separately count (i) sequential batched denoiser calls, (ii) distinct temporal supports, (iii) condition–time branches inside the batch, and (iv) latent updates. ECT uses $(1, 1, 2, 1)$ under this ordering, whereas matched ChordEdit uses $(1, 2, 4, 1)$. Resource values are meaningful only under matched hardware, precision, resolution, batch construction, software stack, synchronization, process scope, and sample order; they should not be interpreted as hardware-independent model properties.

F ImgEdit-Judge Evaluation Protocol

We use the released ImgEdit-Judge checkpoint (Ye et al., 2026), a Qwen2.5-VL-7B evaluator fine-tuned for image-editing assessment, with one fixed three-dimension scoring prompt for all methods. It is loaded in bfloat16 with automatic device placement; Flash Attention 2 is used when available and SDPA otherwise. Decoding is deterministic (`do_sample=False`) with at most 512 new tokens, and the processor uses `min_pixels=1,016,064` and `max_pixels=1,354,752`.

Input and rubric. For sample i, the judge receives the scoring prompt with the edit instruction inserted, followed by the original image and edited image:

$$\mathcal{J}_i = (P(e_i), I_i^{\mathrm{src}}, I_i^{\mathrm{edit}}). \tag{41}$$

It returns one integer in $\{1, 2, 3, 4, 5\}$ for each independent dimension in Table 6. Requested changes are not penalized under CP, and VQ and PC are scored independently of whether the requested edit was completed. CLIP explicitly measures target alignment.

Parsing and aggregation. The requested response contains a brief rationale and the three named scores. We strip Markdown markers and parse either the full dimension names or the fallback

Table 6: ImgEdit-Judge scoring anchors.

Score 1: poor	Score 3: adequate	Score 5: excellent
CP: Content and Identity Preservation		
Severe unintended loss, deformation, or identity/layout change.	Source remains recognizable, with visible unintended local changes.	Unrequested identity, geometry, layout, and detail are preserved.
VQ: Visual Quality and Artifact Control		
Severe blur, noise, corruption, or deformation.	Usable output with visible local blur, noise, or edge artifacts.	Sharp, detailed output without visible editing artifacts.
PC: Physical and Perceptual Coherence		
Severe conflicts in lighting, perspective, anatomy, scale, depth, material, or style.	Broadly plausible, with minor physical or perceptual inconsistency.	Edited content is naturally integrated and internally coherent.

labels `Score 1--3`. A missing or out-of-range value is recorded as missing, never as zero. For dimension q, with valid set V_q, the reported value is

$$\bar{s}_q = \frac{1}{|V_q|} \sum_{i \in V_q} s_{i,q}, \qquad s_{i,q} \in \{1, 2, 3, 4, 5\}. \tag{42}$$

Thus CP, VQ, and PC may have different denominators. We retain each count and list of missing cases, report the three means separately, and do not form an undocumented composite judge score.

G User-Study Protocol

Design. Participants evaluate assigned subsets of PIE-Bench cases; the number of assigned cases need not be identical across participants. Each case contains the source image, edit instruction, and outputs from TurboEdit, SwiftEdit, ChordEdit, and ECT. The comparison uses cases with all four outputs. Each participant rates the four methods on the three criteria below, and the analysis averages the available participant ratings within each case before comparing methods. Consequently, cases have equal weight in the final win rate even when they receive different numbers of ratings.

Blinding and task. The four outputs are presented anonymously, with method identities withheld. Each output receives independent integer scores from 1 (worst) to 5 (best) on the three criteria in Table 7, giving 12 ratings for a completed participant–case record. A participant may assign the same integer score to multiple outputs; the final ranking uses the mean across participants, and no ties for the highest case-level mean occurred in the reported results.

Table 7: Human rating dimensions.

Dimension	Rating criterion
Text alignment	Agreement between the edited image and instruction
Structure preservation	Preservation of non-target content, identity, geometry, and layout
Overall aesthetics	Naturalness, detail, consistency, and artifact control

Case-level aggregation. Let $s_{p,i,m,k} \in \{1, \ldots, 5\}$ be participant p's score for method m on case i and criterion k. We first average across participants within each case,

$$\bar{s}_{i,m,k} = \frac{1}{|\mathcal{P}_i|} \sum_{p \in \mathcal{P}_i} s_{p,i,m,k}, \qquad W_{m,k} = \frac{100}{N} \sum_{i=1}^{N} \mathbf{1}\left[m = \arg\max_{m'} \bar{s}_{i,m',k} \right], \tag{43}$$

where $\mathcal{P}_i$ contains the participants rating case i, and N is the number of evaluated cases. The reported first-place share $W_{m,k}$ is thus a case-level win rate, not a participant-level vote share or a normalized mean rating. No ties for the highest case-level mean were observed, so each case contributes exactly one winner per criterion and the four shares sum to 100%. Figure 6 gives the resulting percentages, rounded to one decimal place. Only aggregate win rates are disclosed.

H Additional Qualitative Examples of the Transport Effect

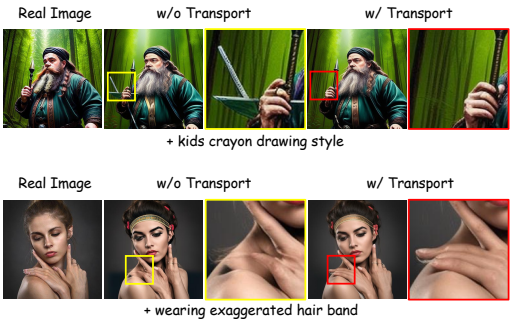


Figure 8: Additional transport ablations complementing Figure 5. From left to right: source, result without transport and its crop, then the "w/ Transport" result and its crop. The highlighted regions show an extra shaft and altered hand–shoulder geometry without transport, reduced with the "w/ Transport" result.

The two cases use PIE-Bench and SD-Turbo under the default single-evaluation setting. The crops highlight an unintended extra shaft in the stylization example and altered hand–shoulder geometry in the hair-band example. The "w/ Transport" result reduces these visible artifacts in the shown cases.

I Additional Qualitative Results

Figure 9 compares ECT, ChordEdit, and FlowEdit on six PIE-Bench cases. The left group contains a glowing-jar-to-crystal-ball edit, stained-glass stylization, and pixel-art stylization. The right group contains a young-man-to-robot edit, flowers-in-mouth addition, and exaggerated-hair-band addition. Each case orders the source, ECT, ChordEdit, and FlowEdit from left to right.

ECT and ChordEdit produce explicit stained-glass boundaries and pixel blocks while retaining much of the source composition. In the robot example, ECT retains more facial appearance while ChordEdit applies a more extensive mechanical transformation. The hair-band case shows the requested accessory in ECT's output together with a broadly preserved hand pose. The flower example also exposes a limitation: ECT adds flowers near the dog's face and suit, but does not clearly place them in its mouth. These cases illustrate the preservation–alignment balance of a single update; retaining composition alone does not establish complete instruction fulfillment.

J Additional Experiments: Seed Stability, Extra-NFE Refinement, and Edit Strength

All experiments here use PIE-Bench and SD-Turbo. The seed and λ studies retain the main single-evaluation editor ($t = 0.75$, $\zeta = 0.7$, $\lambda = 1$, except for the varied parameter). Appendix J.2 instead studies a separate refinement variant that spends an additional model evaluation; its t_{prox} parameter is absent from the main editor. Metrics follow Appendices D–F, with the same display rescaling as the main tables. Blue marks the main default in the seed and λ tables and the best tested extra-NFE setting in Table 9. The matched ChordEdit row is repeated for reference.

J.1 Stability Across Random Seeds

ECT draws a single Gaussian sample ϵ to form the shared noisy state z_t in Equation 4, and this is the only stochastic input to the entire editor: the transport in Equation 12 and the correction in Equation 18 are deterministic functions of the resulting predictions. A concern for single-evaluation editing is dependence on a favorable noise draw. Table 8 evaluates eleven seeds spanning four orders of magnitude on all PIE-Bench cases, with other settings fixed.

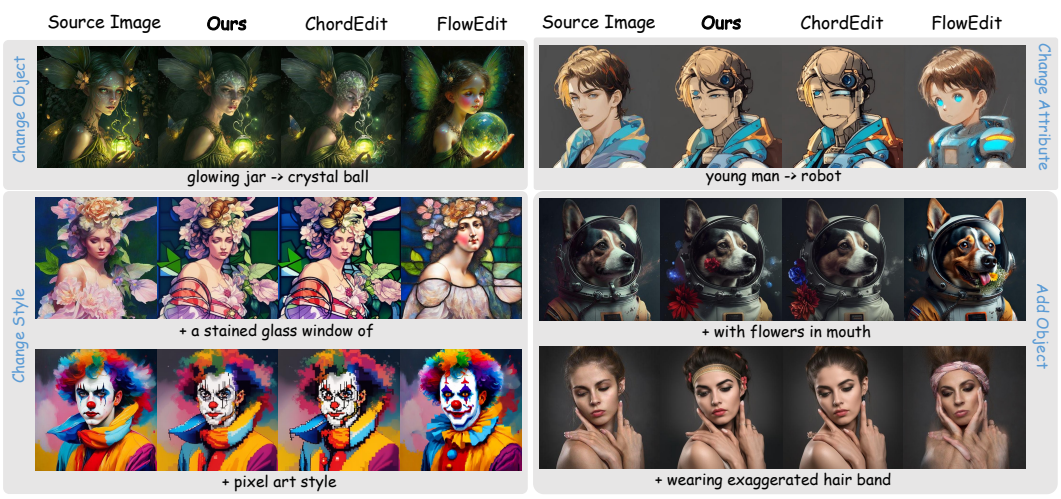


Figure 9: Six qualitative comparisons on PIE-Bench, with source, ECT (ours), ChordEdit, and FlowEdit in each case. Left: object replacement and two style edits. Right: a robot transformation and two object additions. The examples show differences in edit extent and source preservation, including incomplete spatial placement in the flowers-in-mouth case.

Table 8: Seed stability on PIE-Bench (SD-Turbo). Eleven independent seeds, all other settings fixed. The standard deviations across seeds are about an order of magnitude smaller than the ECT–ChordEdit gap on every preservation metric, and the *worst* seed still outperforms matched ChordEdit on all four. The default seed is shaded blue; the reference row is the matched ChordEdit entry of Table 1.

	Consistency (background preservation)				
Seed	PSNR ↑	MSE ↓	LPIPS ↓	SSIM ↑	CLIP ↑
1	23.428	6.641	112.832	77.825	30.23
20	23.598	6.435	110.950	77.834	30.43
64	23.550	6.447	111.065	77.816	30.28
100	23.617	6.452	111.691	77.775	30.34
512	23.548	6.513	112.784	77.888	30.42
1024	23.680	6.354	111.004	77.940	30.34
1536	23.544	6.514	113.927	77.859	30.34
2048	23.595	6.402	112.697	77.948	30.31
4096	23.503	6.630	113.417	77.833	30.28
100000	23.613	6.418	112.478	77.874	30.24
42 (default)	23.740	6.220	109.040	78.100	30.20
Mean ± std	23.583 ±0.084	6.457 ±0.119	111.990 ±1.399	77.881 ±0.089	30.31 ±0.07
Worst seed	23.428	6.641	113.927	77.775	30.20
Ref.: ChordEdit (SD-T)	*22.630*	*7.990*	*122.640*	*76.710*	*30.60*

Takeaway. The seed-induced standard deviation is 0.084 dB PSNR, 0.119 MSE, 1.399 LPIPS, and 0.089 SSIM, i.e. a relative dispersion of 0.36%, 1.85%, 1.25%, and 0.11% of the corresponding mean. These are 9.7–15.6× smaller than the improvement ECT obtains over matched ChordEdit on the same metrics (1.11 PSNR, 1.77 MSE, 13.60 LPIPS, 1.39 SSIM). Every seed, including the worst, exceeds matched ChordEdit on PSNR, MSE, LPIPS, and SSIM; the ranking in Table 1 therefore does not depend on the default draw. CLIP varies by only 0.23 points across seeds, so the modest target-alignment trade-off discussed in Section 4.2 is likewise a property of the method rather than of the sample. This supports editing stability without selecting a favorable random seed.

J.2 Supplementary Refinement with an Extra NFE

Table 3(b) isolates angular transport, which requires no extra evaluation. Panel (c) compares variants without and with an extra neural function evaluation (NFE), solely as an ablation. The ECT results

in Tables 1 and 2 retain one batched denoiser evaluation and one latent update. This supplementary study examines extra-NFE noise-and-denoise refinement; its t_{prox} is not a parameter of the default editor. The latent x_0 supplied to refinement is re-noised and denoised as

$$z_{t_{\text{prox}}} = \alpha_{t_{\text{prox}}} x_0 + \sigma_{t_{\text{prox}}} \epsilon, \qquad \hat{x}_0 = \frac{z_{t_{\text{prox}}} - \sigma_{t_{\text{prox}}} \hat{\epsilon}_\phi(z_{t_{\text{prox}}}, t_{\text{prox}})}{\alpha_{t_{\text{prox}}}}, \tag{44}$$

Table 9 evaluates the extra-NFE refinement in Equation 44 using ImgEdit-Judge. It uses a different PIE-Bench sample set from Table 3(c), precluding direct score comparisons. Its schedule factors describe supplementary noise levels, not the main gain at $t = 0.75$ in Equation 22; its scores concern the refinement variant, not the main-table ECT.

Table 9: Supplementary extra-NFE refinement on PIE-Bench (SD-Turbo). This variant is not used in the main results. Among the tested refinement levels, $t_{\text{prox}} = 0.3$ (blue) has the highest CP, VQ, and PC. ChordEdit is a reference, not an equal-NFE control. Best value per column in bold.

	VP schedule				ImgEdit-Judge		
Setting	α_{prox}	σ_{prox}	$\theta_{\text{prox}}/\sigma_{\text{prox}}$	$\theta_{\text{prox}}/\sigma_{\text{prox}} - \zeta$	CP ↑	VQ ↑	PC ↑
$t_{\text{prox}} = 0.1$	0.945634	0.325234	1.019	0.319	2.92	4.33	3.99
$t_{\text{prox}} = 0.3$	0.768440	0.639921	1.085	0.385	**4.04**	**4.50**	**4.18**
$t_{\text{prox}} = 0.5$	0.525673	0.850686	1.196	0.496	3.48	4.35	3.96
$t_{\text{prox}} = 0.7$	0.288320	0.957534	1.335	0.635	3.30	4.15	3.59
Ref.: ChordEdit (SD-T)	—	—	—	—	*3.95*	*4.37*	*4.00*

Subscript prox denotes evaluation at t_{prox}, with $\theta_{\text{prox}} = \text{atan2}(\sigma_{\text{prox}}, \alpha_{\text{prox}})$ and $\zeta = 0.7$. These are schedule-derived factors; the extra model prediction is not a scalar rescaling of the original residual.

Takeaway. Within this supplementary sweep, $t_{\text{prox}} = 0.3$ gives CP/VQ/PC of $4.04/4.50/4.18$, exceeding the ChordEdit reference on all three dimensions. These scores remain below the main single-evaluation ECT's $4.09/4.65/4.31$ in Table 1. Thus the tested extra refinement does not improve upon the main editor on these judged metrics despite spending an additional NFE. The value 0.3 is only the best tested setting of this variant, not a parameter of the main method.

J.3 Edit-Strength Coefficient λ

Table 10 varies $\lambda \in \{1, \ldots, 6\}$ with all other settings fixed. This coefficient scales the full update in Equation 22, preserving its direction without another evaluation.

Table 10: Effect of the edit-strength coefficient λ on PIE-Bench (SD-Turbo). Preservation degrades monotonically with λ; CLIP peaks at $\lambda = 2$ and then declines. Its 0.50-point gain costs 2.690 dB PSNR and 37.636 scaled LPIPS units. Blue marks $\lambda = 1$, the preservation-oriented default.

	Consistency (background preservation)				
Setting	PSNR ↑	MSE ↓	LPIPS ↓	SSIM ↑	CLIP ↑
$\lambda = 1$ (ours)	**23.740**	**6.220**	**109.040**	**78.100**	30.20
$\lambda = 2$	21.050	11.655	146.676	74.432	**30.70**
$\lambda = 3$	19.084	18.150	183.486	71.209	30.60
$\lambda = 4$	17.724	24.364	213.623	68.508	30.20
$\lambda = 5$	16.717	30.136	239.572	66.179	29.90
$\lambda = 6$	15.933	35.463	263.881	64.077	29.70
Ref.: ChordEdit (SD-T)	*22.630*	*7.990*	*122.640*	*76.710*	*30.60*

Takeaway. CLIP peaks at $\lambda = 2$ and falls below its initial value for $\lambda \geq 5$. From $\lambda = 1$ to 6, PSNR loses 7.807 dB and LPIPS grows by 154.841. These are empirical sensitivity results: λ scales the final update while the angular horizon stays fixed. We default to $\lambda = 1$, the only tested value improving all four preservation metrics over matched ChordEdit; users can adjust λ for edit strength.