VINS-DISTILL: SOURCE-PRESERVING FEW-STEP IMAGE EDITING WITH PROMPT-REWRITE-AUGMENTED REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Few-step distillation for image editing requires reducing inference cost while maintaining editing capability and preserving fine details of the source image. However, editing teachers may alter content beyond the requested edit, and few-step distillation can inherit and amplify these errors. To address this problem, we introduce VINS-Distill, which uses source-anchored target projection (SATP) to correct unintended changes in the teacher's distillation targets and prompt-rewrite-augmented reinforcement learning (PRA-RL) to strengthen editing capability. Specifically, SATP replaces teacher denoising targets in non-edited regions with source-derived latent values. PRA-RL further strengthens editing capability by injecting rewritten instructions during denoising while retaining the original instruction for reward evaluation. Together, SATP and PRA-RL enable the four-step editor to preserve source content while maintaining strong editing capability, without requiring rewritten instructions at inference. To comprehensively evaluate the source preservation and editing quality of our method, we introduce VINS-4K-Bench, comprising 3,305 tasks across diverse source images and edit categories to address the scarcity and limited diversity of existing 4K editing benchmarks. Evaluation on VINS-4K-Bench and two established 1K benchmarks shows that our four-step editor consistently improves source preservation and editing quality over existing state-of-the-art few-step baselines.

1 INTRODUCTION

Image editing aims to carry out a requested change while preserving the fine textures and local structures of the source image. Advances in latent diffusion and flow-based generation (Rombach et al., 2022; Podell et al., 2024; Esser et al., 2024) have enabled models that perform such edits from natural-language instructions (Brooks et al., 2023; Labs et al., 2025; Labs, 2025). However, repeated denoising remains computationally costly, especially at high resolutions such as 4K. Reducing this sampling cost requires few-step editors that retain source details while executing the intended change.

Distillation offers a way to compress iterative sampling into a few steps (Salimans & Ho, 2022; Song et al., 2023; Luo et al., 2023). Distribution matching methods, including DMD, DMD2, and SenseFlow, transfer pretrained generative capabilities to few-step students (Yin et al., 2024b;a; Ge et al., 2025). Recent methods further combine distillation with reinforcement learning (RL) to improve students through reward-based optimization (Jiang et al., 2026; Huang et al., 2026; Luo et al., 2026). When applied to editing, however, teacher-guided distillation can inherit the teacher's preservation errors. A teacher may perform the requested modification while altering unrelated source content (Qian et al., 2025), as illustrated in Figure 1(a). Matching its predictions can therefore transfer and amplify unwanted changes alongside editing ability in the few-step student..

Because the content to be preserved is already present in the source image, distillation has access to a reference beyond the teacher's prediction. This motivates **combining teacher guidance for editing with source-based supervision for preservation**. For 4K editing, this complementary supervision is particularly relevant to fine textures and local structures that should survive the edit.

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

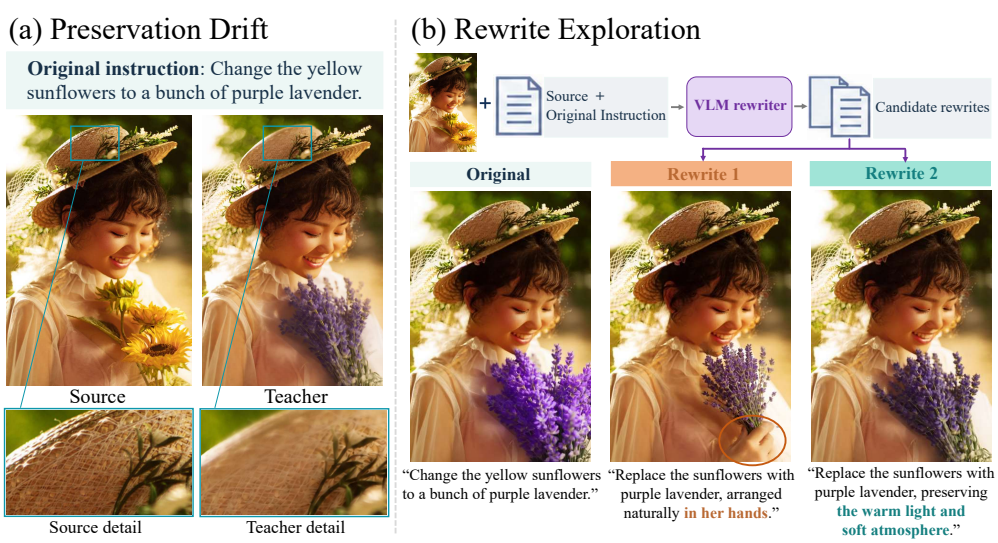

Figure 1: **Motivation.** (a) Preservation drift: the intended edit alters unrelated source content. (b) Rewrite exploration: alternative instructions diversify few-step editing. These motivate source-based preservation and prompt-rewrite-augmented RL.

Alongside source preservation, few-step distillation must retain strong editing capability. Compressing denoising into only a few steps can weaken this capability, motivating RL to improve the distilled student. However, limited stochastic variation in few-step rollouts restricts effective RL exploration. We therefore use source-conditioned instruction rewrites to increase conditioning diversity, while evaluating the resulting edits against the original request.

Building on these observations, we introduce VINS-Distill, a joint distillation and RL framework for source-preserving few-step image editing. *Source-Anchored Target Projection* (SATP) replaces teacher targets in non-edited regions with source-derived latent values for preservation, while *Prompt-Rewrite-Augmented Reinforcement Learning* (PRA-RL) uses source-conditioned rewrites to diversify RL exploration, with rewards based on the original request. Together, they train a single editor that performs four-step inference using only the source image and original instruction.

We evaluate VINS-Distill at both 4K and 1K. For 4K, we train a base editor from FLUX.2-klein-base-9B on quality-filtered data and apply VINS-Distill to obtain a four-step model. To address the scarcity and limited diversity of existing 4K benchmarks, we introduce VINS-4K-Bench, extending VINS-120K (Chen et al., 2026) to 3,305 diverse 4K editing tasks. We also evaluate VINS-Distill at 1K on GEdit-Bench and ImgEdit-Bench (Liu et al., 2025b; Ye et al., 2025). Across all three benchmarks, VINS-Distill improves source preservation and editing quality over existing few-step baselines and achieves the highest overall scores. Our main contributions are:

- We introduce VINS-Distill, a joint distillation and reinforcement learning framework for source-preserving few-step image editing.
- We develop SATP to correct teacher preservation errors using source-derived latent values, and PRA-RL to diversify RL exploration through prompt rewriting while retaining original-task rewards.
- We introduce VINS-4K-Bench, a 3,305-task benchmark addressing the scarcity and limited diversity of existing 4K editing benchmarks, and demonstrate improved source preservation and editing quality across 4K and two established 1K benchmarks.

2 Related Work

Diffusion models for image editing. Early diffusion editors adapt pretrained generators through noisy-image initialization, spatial constraints, or attention control (Meng et al., 2022; Couairon et al., 2023; Hertz et al., 2023; Cao et al., 2023). InstructPix2Pix learns instruction-based editing

from paired data (Brooks et al., 2023), while FLUX.1 Kontext and FLUX.2-klein unify generation and editing through image and text conditioning (Labs et al., 2025; Labs, 2025). Recent work extends editing to ultra-high resolutions (Chen et al., 2026; Song et al., 2026). However, instruction-following models can still alter unrelated content (Qian et al., 2025). We focus on retaining fine source details while distilling these models into few-step UHR editors.

Distribution matching distillation. DMD trains few-step students by matching teacher distributions (Yin et al., 2024b). DMD2 improves training stability and supports multistep sampling, while SenseFlow extends distribution matching to large flow models (Yin et al., 2024a; Ge et al., 2025). DMDR and RTDMD combine distillation with reinforcement learning (Jiang et al., 2026; Huang et al., 2026); TDM-R1 incorporates non-differentiable rewards through learned surrogates (Luo et al., 2026). However, teacher-guided distillation can transfer preservation errors along with editing ability. Our SATP supplies source references within the distillation target, and PRA-RL complements this supervision with reward-based editing refinement.

3 Source Preservation and Prompt-Rewrite-Augmented Reinforcement Learning

Few-step image editing requires both faithful source preservation and strong editing capability. Distillation can transfer the teacher's unwanted changes to source content, while compressing its denoising process can weaken the student's ability to execute edits. Source-Anchored Target Projection (SATP) addresses the first issue by incorporating source references into the teacher's denoising targets. Prompt-Rewrite-Augmented Reinforcement Learning (PRA-RL) addresses the second by augmenting rollout exploration with rewritten instructions and optimizing the student against the original task. Section 4 combines these components in joint training.

Let I_s be the source image, c_0 its editing instruction, $S \in \mathbb{R}^{C\times H\times W}$ its normalized latent, and $h = (S, c_0)$ the editing condition. The generator produces a clean latent $x_\theta = G_\theta(z; h)$, where $z \sim \mathcal{N}(0, I)$ collects its sampling noise and $D = CHW$ is the number of latent coordinates.

3.1 Source-Anchored Target Projection

As shown in Figure 1(a), teacher-guided distillation can transfer unintended source changes to the student. Since the source provides the content to be preserved, SATP uses teacher–source agreement to anchor source values in the denoising target while retaining teacher guidance elsewhere.

Denoising targets. DMD compares teacher and student score estimates at noisy versions of the student's output (Yin et al., 2024b;a). For flow-matching models (Lipman et al., 2023; Liu et al., 2023), we re-noise x_θ at a sampled time t using linear interpolation:

$$x_t = \alpha_t x_\theta + \sigma_t \epsilon, \qquad \alpha_t = 1 - \sigma_t, \quad \epsilon \sim \mathcal{N}(0, I), \tag{1}$$

where ϵ is independent of the generator noise z. A frozen teacher v_T and a trainable fake-score estimator v_ϕ provide the clean estimates

$$\begin{aligned} T_t &= x_t - \sigma_t v_T(x_t, \sigma_t; h), \\ F_t &= x_t - \sigma_t v_\phi(x_t, \sigma_t; h). \end{aligned} \tag{2}$$

The score difference can then be expressed through $F_t - T_t$, yielding the DMD signal in clean-sample space; the derivation appears in Appendix A.1. The fake estimator tracks the student's distribution by fitting detached student outputs; its auxiliary training objective is given in Equation 19 in Appendix A.1.. SATP modifies the teacher reference T_t, which shares the source latent's normalization and spatial layout.

Selecting source anchors. Anchoring every coordinate to the source would suppress the requested edit. We instead use local teacher–source agreement to identify candidate preservation anchors. Where the teacher prediction remains close to the source, source values provide a reference for removing residual deviations; larger differences retain the teacher's guidance. We measure the discrepancy between T_t and S at each spatial location u relative to the source energy:

$$r_t(u) = \frac{C^{-1}\sum_a (T_{t,a,u} - S_{a,u})^2}{C^{-1}\sum_a S_{a,u}^2 + \varepsilon_m}, \qquad M_t(u) = \mathbf{1}[\exp(-\gamma r_t(u)) > \tau]. \tag{3}$$

Here a indexes channels, and the binary spatial mask M_t is broadcast across them. The scale γ and threshold τ control anchor selection, while ε_m stabilizes the denominator. This rule selects low-discrepancy locations separately for each teacher prediction, without annotated edit regions. Parameter settings are given in Appendix A.1.

Projecting the teacher target. Once M_t is selected, the source constraints define the affine set $\mathcal{A}_{M_t}(S) = \{X : M_t \odot X = M_t \odot S\}$. We construct the target by hard coordinate replacement:

$$T_t^{\mathrm{SA}} = (1 - M_t) \odot T_t + M_t \odot S, \tag{4}$$

$$= \operatorname*{argmin}_{X \in \mathcal{A}_{M_t}(S)} \frac{1}{2} \|X - T_t\|_2^2. \tag{5}$$

For the selected mask, this replacement is the Euclidean projection onto $\mathcal{A}_{M_t}(S)$: it makes the smallest change to the teacher target that satisfies the source constraints. Thus, preservation supervision enters through the distillation reference, while unselected coordinates retain the teacher's denoising prediction. Appendix A.2 proves the projection identity and shows that target error decreases when the source is closer to the ideal edit than the teacher on the selected coordinates.

3.2 Prompt-Rewrite-Augmented Reinforcement Learning

Alongside preserving source content, a few-step editor must retain the editing capability of its teacher. Compressing the denoising process into only a few steps can weaken this capability, motivating reinforcement learning to further improve the distilled student. However, under a fixed instruction and a short denoising trajectory, stochastic variation across rollouts can provide limited exploration for reward-based optimization. We therefore introduce source-conditioned instruction rewrites to provide additional conditioning diversity during training, building on language-guided editing and prompt augmentation (Fu et al., 2024; Wang et al., 2026a).

PRA-RL combines this instruction diversity with sampling noise to generate alternative editing trajectories. Rewrites augment exploration while the original request defines the editing objective: all resulting edits are evaluated against that request. This distinction matters because different rewrites can lead to both useful and unwanted changes (Figure 1(b)). We introduce each rewrite at a selected transition and pair noise across instructions to organize the rollout group. GRPO (Shao et al., 2024; Liu et al., 2025a) then uses original-task rewards to update the editor at that transition.

Instruction rewriter. We train a source-conditioned rewriter q_ψ on instruction-expansion examples, each containing a source image, a concise request, and a detailed target instruction. It learns to make the requested changes and preservation requirements explicit in the context of the source. With a fixed editor, its rewrites improve overall editing scores and success rates on all three benchmarks (Table 6). These results motivate using the rewriter to supply additional conditioning choices for RL exploration. Model and training details are provided in Appendix A.5. We freeze q_ψ and, given I_s and c_0, obtain $J - 1$ rewritten instructions. Together with the original instruction, they form

$$\mathcal{C}(I_s, c_0) = \{c_0, c_1, \ldots, c_{J-1}\}, \qquad c_j \sim q_\psi(\cdot \mid I_s, c_0) \quad (j > 0). \tag{6}$$

All candidates remain fixed while we collect and optimize the task's rollout group.

Step-local prompt rewriting. The generator samples a trajectory $y_0, \ldots, y_K$ along noise levels $1 = \sigma_0 > \cdots > \sigma_K = 0$, starting from a Gaussian latent and ending at a clean output. Here k indexes sampling steps, distinct from the distillation time t in Equation 1. Following Flow-GRPO (Liu et al., 2025a), each stochastic transition is a Gaussian policy $\pi_\theta^{(k)}(\cdot \mid y_k, S, c) = \mathcal{N}(\mu_{\theta,k}(y_k; S, c), \nu_k^2 I)$. Its mean is computed from the velocity network v_θ underlying G_θ and the sampling schedule. The transition variance remains fixed during optimization, and the last step is deterministic ($\nu_{K-1} = 0$). The generator noise z collects the initial and transition noises.

We introduce conditioning diversity at one stochastic transition, allowing the student to explore alternative denoising directions while using the original instruction at the remaining steps. For each rollout group, we sample $k^\star \sim \mathrm{Unif}\{0, \ldots, K-2\}$. Candidate j supplies the instruction only at this transition:

$$c_j^{(k)} = \begin{cases} c_j, & k = k^\star, \\ c_0, & k \neq k^\star. \end{cases} \tag{7}$$

With collection parameters θ_{old} fixed, we generate N rollouts for each candidate:

$$y_{j,i,k+1} = \mu_{\theta_{\text{old}},k}(y_{j,i,k}; S, c_j^{(k)}) + \nu_k \xi_{j,i,k}, \qquad i = 1, \ldots, N. \tag{8}$$

At the selected transition, different instructions condition the policy mean, complementing the variation introduced by Gaussian noise. Their effects propagate through the remaining original-instruction steps to the final edit, whose reward provides feedback for learning.

Pairing noise across instructions. To combine instruction and noise variation in the rollout group, all JN trajectories share the initial latent y_0. At $k^\star$, repetition i uses an independent noise draw shared across candidate instructions; the other stochastic transitions share noise across the entire task group:

$$\xi_{j,i,k} = \begin{cases} \xi_{i,k^\star}, & k = k^\star, \\ \xi_k, & k \neq k^\star, \end{cases} \qquad \xi_{i,k^\star}, \xi_k \sim \mathcal{N}(0, I). \tag{9}$$

The resulting $J \times N$ grid explores both conditioning choices across rows and selected-step noise across columns. Within each column, candidate instructions are evaluated under the same noise draws; across columns, independent draws provide stochastic variation at $k^\star$. All trajectories share the prefix before this step. This allocation organizes the additional conditioning diversity into paired rollouts, whose final rewards enter a common optimization group.

Original-task feedback. We use the frozen MoRE-OPD model (Anonymous, 2027) to provide a reward for each completed edit, conditioned on the source image and original instruction:

$$R_{j,i} = R(\mathcal{D}(y_{j,i,K}); I_s, c_0), \tag{10}$$

where $\mathcal{D}$ is the frozen decoder. All JN rewards form one original-task group, so original and rewritten conditions contribute to the same group-relative feedback. Rewrites change the conditions explored during training, while c_0 remains the criterion for editing success and source preservation. GRPO uses this feedback to update only the selected transition; the rewriter remains frozen. The original-instruction trajectories also supply samples for SATP, allowing the two components to jointly train the editor. At inference, it uses the original instruction without requiring rewrites. The joint objectives are described next.

4 The VINS-Distill Training Framework

In this section, we detail the overall framework of VINS-Distill, as illustrated in Figure 2.

4.1 Stage I: Cold-Start Initialization

We adopt RTDMD's cold-start distillation (Huang et al., 2026) under $h = (S, c_0)$ to initialize the generator and fake-score estimator. Training settings are given in Appendix A.4.

4.2 Stage II: Joint Refinement

In the $J \times N$ rollout group from Section 3.2, the original-instruction row ($j = 0$) supplies SATP samples and all rows supply GRPO rewards. The shared editor minimizes

$$\mathcal{L}_\theta = \lambda_D \mathbb{E}[\mathcal{L}_{\text{SATP}}] + \lambda_R \mathcal{L}_{\text{PRA-RL}}. \tag{11}$$

The expectation averages original-row repetitions and distillation noise, while λ_D and λ_R weight the branches. Rollouts are collected under fixed θ_{old} and cached without gradients.

Source-anchored distillation. For each original-instruction rollout, we recompute the final clean prediction x_θ from its cached input. Re-noising it yields the teacher and fake estimates, and SATP gives

$$\begin{aligned} g_{\text{SA}} &= F_t - T_t^{\text{SA}} \\ &= (1 - M_t) \odot (F_t - T_t) + M_t \odot (F_t - S). \end{aligned} \tag{12}$$

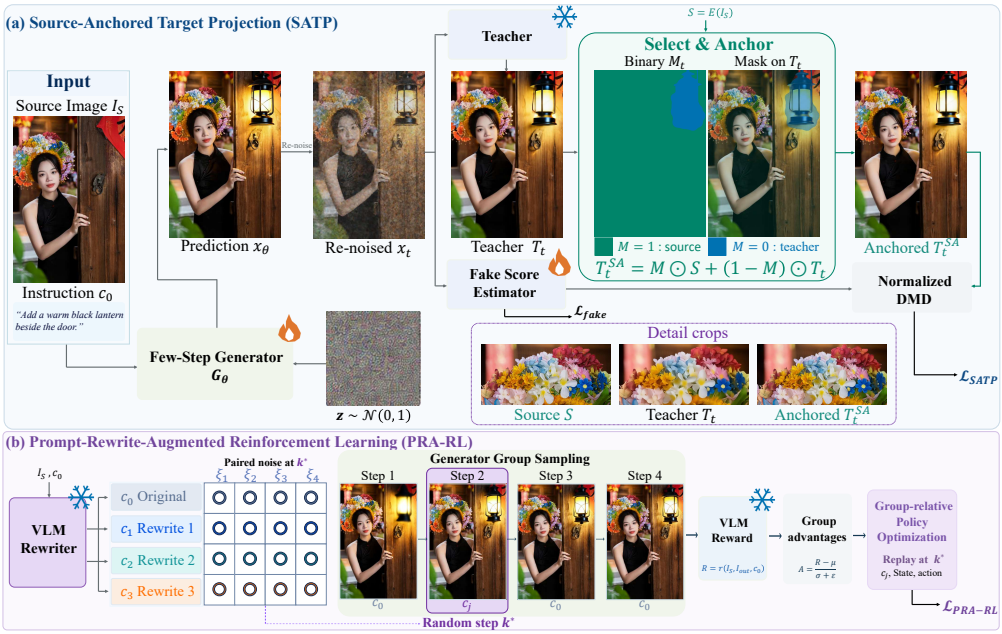

Figure 2: Overview of **VINS-Distill**. (a) SATP anchors teacher targets to the source. (b) PRA-RL uses paired-noise, step-local rewrites for GRPO. Both jointly train one editor in latent space.

We normalize this signal by the mean absolute residual and use a stop-gradient surrogate:

$$n_{\mathrm{SA}} = D^{-1}\|x_\theta - T_t^{\mathrm{SA}}\|_1, \qquad \bar{g}_{\mathrm{SA}} = g_{\mathrm{SA}}/n_{\mathrm{SA}},$$
$$\mathcal{L}_{\mathrm{SATP}} = \frac{1}{2D}\,\|x_\theta - \mathrm{sg}[x_\theta - \bar{g}_{\mathrm{SA}}]\|_2^2\,. \tag{13}$$

Here sg stops gradients through the target; gradients pass only through the final prediction recomputed from the detached cached state. The fake estimator separately fits detached original-instruction outputs using Equation 19; this auxiliary update supplies no gradient to the editor.

GRPO at the selected transition. Terminal rewards yield group-relative advantages:

$$A_{j,i} = \mathrm{sg}\left[\frac{R_{j,i} - \bar{R}}{s_R + \epsilon_A}\right], \tag{14}$$

where $\bar{R}$ and s_R are the mean and population standard deviation of all JN rewards. With clipping threshold δ, the selected transition uses clipped GRPO (Schulman et al., 2017; Shao et al., 2024; Liu et al., 2025a):

$$\mathcal{L}_{\text{PRA-RL}} = -\frac{1}{JN}\sum_{j,i}\min\left\{\rho_{j,i}(\theta)A_{j,i},\ \mathrm{clip}\big(\rho_{j,i}(\theta), 1-\delta, 1+\delta\big)A_{j,i}\right\}. \tag{15}$$

The ratio $\rho_{j,i}$ compares current and collection policies at $k^\star$ under (S, c_j), using log-probabilities averaged over latent coordinates. GRPO differentiates only this transition, with cached states and actions held fixed (Appendix A.3).

Both objectives update the shared editor parameters θ; the teacher, rewriter, evaluator, and autoencoder remain frozen. Appendix A.4 gives the update sequence. At inference, the editor uses the original instruction throughout four-step sampling.

5 Experiments

We evaluate whether VINS-Distill can preserve source details and carry out requested edits with four-step sampling at 4K resolution. We then assess its applicability at 1K on GEdit-Bench and ImgEdit-Bench.

Table 1: **Editing quality on VINS-4K-Bench.** Best and second-best values are **bold** and underlined.

	MoRE-OPD					Additional Reward	
Method	IF ↑	NC ↑	VQ ↑	Score ↑	SR (%) ↑	SpatialReward ↑	RationalRewards ↑
4K base editor	7.7841	7.4915	7.3834	7.2037	36.67	7.0386	0.7240
TDM-R1	8.4119	8.4319	8.4476	8.1366	52.22	7.4618	0.8030
RTDMD	8.3766	8.3163	8.4753	8.0999	50.68	7.5518	0.7973
VINS-Distill	**8.5386**	**9.0506**	**8.7802**	**8.4831**	**60.79**	**7.6784**	**0.8071**

5.1 Experimental Setup

Models and training. Our 4K model adapts FLUX.2-klein-base-9B through latent-space adaptation and progressive 1K-to-4K fine-tuning on quality-filtered data; 1K experiments use the base model directly. Following RTDMD, we jointly refine the editor with SATP and PRA-RL, using the original instruction, three rewrites, and four noise repetitions each. MoRE-OPD (Anonymous, 2027) provides the RL reward. The resulting editor performs 4 step inference using only the source image and original instruction. Appendices B.1 and A.4 provide training details.

Benchmarks and evaluation. VINS-4K-Bench extends VINS-120K (Chen et al., 2026) with diverse native-4K images and manually curated instructions across nine editing categories. At 1K, we evaluate on GEdit-Bench and ImgEdit-Bench (Liu et al., 2025b; Ye et al., 2025). MoRE-OPD (Anonymous, 2027) scores instruction following (IF), non-edited content preservation (NC), visual quality (VQ), overall Score, and success rate (SR). We additionally report SpatialReward (Long et al., 2026) and RationalRewards (Wang et al., 2026b). All metrics evaluate the original editing task and are higher-is-better. Appendix B.2 describes VINS-4K-Bench.

5.2 Ultra-High-Resolution Editing at 4K

Table 1 compares VINS-Distill with TDM-R1 and RTDMD using the same 4K backbone and four-step sampling, with the 50-step base editor as reference. VINS-Distill achieves the best results on all metrics. Compared with other four-step methods, it substantially improves non-edited content preservation while achieving stronger instruction following and visual quality. The gains in overall Score and success rate indicate that source preservation is compatible with effective editing, which is central to retaining fine details in UHR images. SpatialReward and RationalRewards also rank VINS-Distill highest, supporting the comparison beyond the training evaluator.

As illustrated in Figure 3, VINS-Distill performs the requested changes while retaining fine structure and texture in unedited regions. The detail views make preservation differences visible that are less apparent when viewing the full 4K images. These observations complement the quantitative results and motivate using the source image to anchor distillation targets in regions that should remain unchanged. Appendix B.5 reports the inference-time comparison.

5.3 Additional Evaluation at 1K

Table 2 compares VINS-Distill with DMD, DMD2, SenseFlow, TDM-R1, RTDMD, and the official Klein models at 1K. VINS-Distill achieves the best results on all seven metrics on GEdit-Bench. On ImgEdit-Bench, it leads in all metrics except IF, where Klein-9B performs best. Across both benchmarks, VINS-Distill outperforms the compared distillation methods on every metric, demonstrating that the benefits of source-anchored distillation and reward-based refinement extend beyond the 4K setting. Appendix B.6 provides visual comparisons.

5.4 Ablation Studies

All ablations use VINS-4K-Bench with four-step inference and matched settings, while keeping the other component fixed.

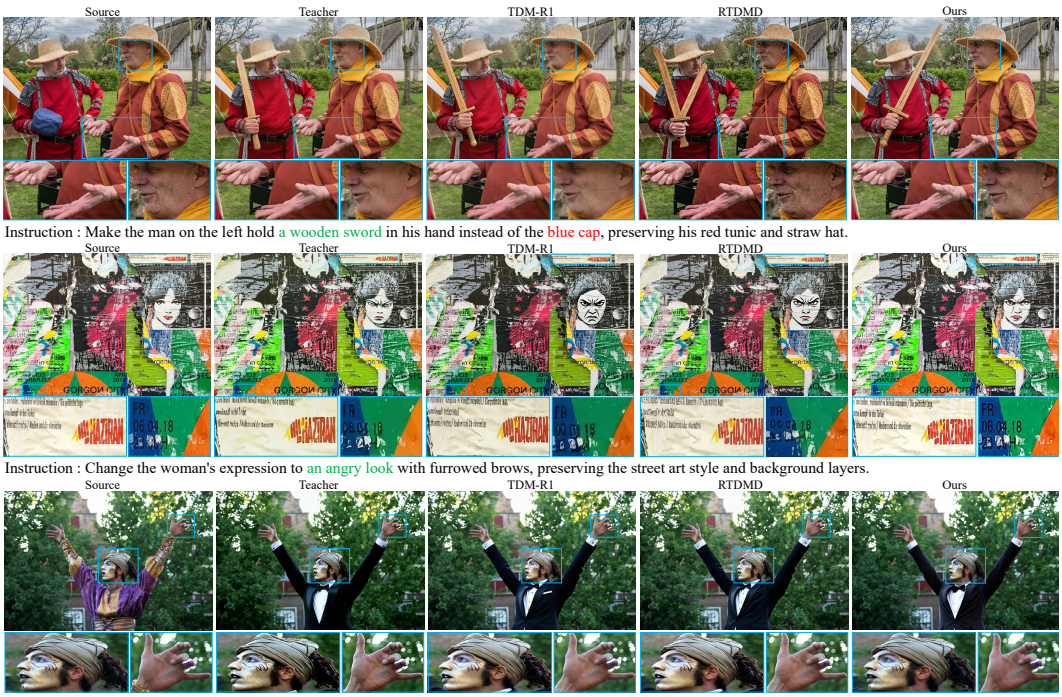

Figure 3: **Qualitative comparison at 4K.** Detail views highlight source preservation alongside the requested edits.

Table 2: **Editing quality at 1K.** Best and second-best values are **bold** and underlined.

Benchmark	Method	MoRE-OPD					Additional Reward	
		IF ↑	NC ↑	VQ ↑	Score ↑	SR (%) ↑	SpatialReward ↑	RationalRewards ↑
GEdit-Bench	Klein-base-9B	8.1860	8.5540	8.8020	7.9680	51.16	7.0724	0.7522
	Klein-9B	8.3640	8.6260	8.8460	8.1040	51.57	7.3234	0.7638
	DMD	7.8880	8.7610	8.7820	7.8630	50.50	6.9514	0.7459
	DMD2	7.9870	8.6500	8.7820	7.9110	50.83	7.1097	0.7497
	SenseFlow	8.0070	8.5510	8.7440	7.8950	49.67	7.1134	0.7588
	TDM-R1	8.3120	8.6290	8.8134	8.1125	52.81	7.2964	0.7647
	RTDMD	8.3651	8.6348	8.8274	8.0927	51.82	7.3144	0.7668
	VINS-Distill	**8.4257**	**8.7624**	**8.9026**	**8.1997**	**55.28**	**7.4130**	**0.7773**
ImgEdit-Bench	Klein-base-9B	7.6280	8.8220	8.8740	7.6050	48.03	6.7957	0.7765
	Klein-9B	**7.8960**	8.8170	8.8670	7.8170	48.54	6.9829	0.7781
	DMD	7.3950	8.7220	8.6200	7.4300	44.85	6.7342	0.7598
	DMD2	7.5950	8.7870	8.7290	7.5900	45.11	6.8542	0.7721
	SenseFlow	6.6630	8.5130	8.4640	6.8650	33.93	6.1524	0.7335
	TDM-R1	7.7892	8.8145	8.8612	7.8020	51.33	6.9420	0.7712
	RTDMD	7.7923	8.8250	8.8721	7.8124	51.08	6.9471	0.7794
	VINS-Distill	7.8653	**8.9898**	**9.0114**	**7.8562**	**53.37**	**7.0130**	**0.7874**

Component complementarity. Table 3 compares SATP, PRA-RL, and their joint training against the cold start. SATP improves preservation and visual quality with a small IF decrease. PRA-RL improves IF but reduces NC and VQ. Their joint use achieves the best IF, VQ, and overall Score while retaining most of the preservation gain from SATP. Although SATP alone achieves the highest NC and success rate, the joint results confirm their complementary roles in preservation and editing.

Table 3: **Framework ablation on VINS-4K-Bench.**

	MoRE-OPD					Additional Reward	
Variant	IF ↑	NC ↑	VQ ↑	Score ↑	SR (%) ↑	SpatialReward ↑	RationalRewards ↑
Cold Start	8.2436	8.5876	8.5739	8.1268	54.25	7.5213	0.7891
w/ SATP	8.1742	**9.1842**	8.7564	8.4232	**62.75**	7.6385	0.8061
w/ PRA-RL	8.4586	8.4213	8.3984	8.1436	55.95	7.5361	0.7886
w/ SATP + PRA-RL (ours)	**8.5386**	9.0506	**8.7802**	**8.4831**	60.79	**7.6784**	**0.8071**

Table 4: **SATP ablation on VINS-4K-Bench.**

	MoRE-OPD					Additional Reward	
Distillation variant	IF ↑	NC ↑	VQ ↑	Score ↑	SR (%) ↑	SpatialReward ↑	RationalRewards ↑
DMD	8.2845	8.4751	8.4126	8.1053	49.68	7.5196	0.7865
DMD + masked source L_2	8.2184	8.6156	8.4352	8.1923	53.92	7.5435	0.7937
SATP (ours)	**8.5386**	**9.0506**	**8.7802**	**8.4831**	**60.79**	**7.6784**	**0.8071**

Table 5: **PRA-RL ablation on VINS-4K-Bench.**

	MoRE-OPD					Additional Reward	
Prompt conditioning	IF ↑	NC ↑	VQ ↑	Score ↑	SR (%) ↑	SpatialReward ↑	RationalRewards ↑
w/o prompt rewriting	8.3641	8.7251	8.6420	8.4452	56.01	7.6528	0.8067
Full-trajectory prompt rewriting	8.4412	8.8817	8.6915	8.4315	54.83	7.6619	0.8033
PRA-RL (ours)	**8.5386**	**9.0506**	**8.7802**	**8.4831**	**60.79**	**7.6784**	**0.8071**

Source supervision. With PRA-RL fixed, Table 4 compares SATP with DMD and DMD with a masked source L_2 loss. The latter improves preservation but reduces IF, revealing a trade-off from aligning the student output with the source. SATP achieves the best results across all metrics by anchoring selected teacher targets to the source while retaining teacher guidance elsewhere.

Prompt-conditioning scope. With SATP fixed, Table 5 compares original-instruction GRPO, full-trajectory rewriting, and step-local rewriting. All use 16 rollouts and update one selected transition. PRA-RL achieves the best results across all metrics. Full-trajectory rewriting improves IF, NC, and VQ but lowers overall Score and success rate, showing that persistent rewriting is not consistently beneficial. These results favor step-local rewriting while retaining the original instruction elsewhere. Appendix B.4 provides variant definitions and training curves.

6 Conclusion

We presented VINS-Distill for source-preserving few-step image editing. SATP corrects unintended changes in teacher distillation targets using source-derived latent values, while PRA-RL strengthens editing through prompt rewrites evaluated against the original request. We also introduced VINS-4K-Bench with 3,305 diverse 4K editing tasks for evaluating source preservation and editing quality. Experiments on VINS-4K-Bench and two 1K benchmarks demonstrate improved source preservation and editing quality over existing few-step baselines.

Limitations. SATP may misidentify anchors for subtle edits or large teacher deviations, and its latent-space supervision depends on the autoencoder's ability to preserve fine details. PRA-RL depends on the rewriter and reward model and introduces additional training cost. Improving anchor selection and training efficiency remains future work.

AI USE STATEMENT

In this work, we used generative AI tools to generate training data and assist in the construction of VINS-4K-Bench, as described in the paper. We did not use generative AI tools to develop theoretical models, formulate mathematical claims, assist with proofs, or conduct qualitative or thematic data analysis. Additionally, we used generative AI tools to draft parts of the manuscript and to edit the language for improved readability. All AI-generated data and AI-assisted benchmark construction were subject to the quality control and evaluation procedures described in the paper, and all AI-assisted manuscript content was reviewed by the authors. We take responsibility for the final content of this work, including text, claims, data, and artifacts produced with the aid of generative AI.

REFERENCES

Anonymous. MoRE-OPD: Mixture of reward experts with on-policy distillation for image editing, 2027. Under review.

Samyadeep Basu, Mehrdad Saberi, Shweta Bhardwaj, Atoosa Malemir Chegini, Daniela Massiceti, Maziar Sanjabi, Shell Xu Hu, and Soheil Feizi. Editval: Benchmarking diffusion based text-guided image editing methods. *CoRR*, abs/2310.02426, 2023. doi: 10.48550/ARXIV.2310.02426. URL https://doi.org/10.48550/arXiv.2310.02426.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 18392–18402. IEEE, 2023. doi: 10.1109/CVPR52729.2023.01764. URL https://doi.org/10.1109/CVPR52729.2023.01764.

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, Paris, France, October 1-6, 2023*, pp. 22503–22513. IEEE, 2023. doi: 10.1109/ICCV51070.2023.02062. URL https://doi.org/10.1109/ICCV51070.2023.02062.

Zhizhou Chen, Shanyan Guan, Zhanxin Gao, En Ci, Yanhao Ge, Wei Li, Zhenyu Zhang, Jian Yang, and Ying Tai. VINS-120K: ultra high-resolution image editing with A large-scale dataset. *CoRR*, abs/2605.23518, 2026. doi: 10.48550/ARXIV.2605.23518. URL https://doi.org/10.48550/arXiv.2605.23518.

Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. 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=3lge0p5o-M-.

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 Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *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.

Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. 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=S1RKWSyZ2Y.

Xingtong Ge, Xin Zhang, Tongda Xu, Yi Zhang, Xinjie Zhang, Yan Wang, and Jun Zhang. Senseflow: Scaling distribution matching for flow-based text-to-image distillation. *CoRR*, abs/2506.00523, 2025. doi: 10.48550/ARXIV.2506.00523. URL https://doi.org/10.48550/arXiv.2506.00523.

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. URL https://doi.org/10.48550/arXiv.2207.12598.

Yushi Huang, Xiangxin Zhou, Ruoyu Wang, Chi Zhang, Jun Zhang, and Tianyu Pang. Reinforcing few-step generators via reward-tilted distribution matching. *CoRR*, abs/2605.26108, 2026. doi: 10.48550/ARXIV.2605.26108. URL https://doi.org/10.48550/arXiv.2605.26108.

Dengyang Jiang, Dongyang Liu, Zanyi Wang, Qilong Wu, Liuzhuozheng Li, Hengzhuang Li, Xin Jin, Changsheng Lu, Zhen Li, Mengmeng Wang, Steven C. H. Hoi, Peng Gao, and Harry Yang. Distribution matching distillation meets reinforcement learning. In Paolo Favaro, Zuzana Kukelova, Atsuto Maki, Anna Rohrbach, Konrad Schindler, and Federico Tombari (eds.), *Computer Vision - ECCV 2026 - 19th European Conference, Malmö, Sweden, September 8-12, 2026, Proceedings, Part XIX*, volume 17019 of *Lecture Notes in Computer Science*, pp. 281–299. Springer, 2026. doi: 10.1007/978-3-032-37550-6_16. URL https://doi.org/10.1007/978-3-032-37550-6_16.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/73aacd8b3b05b4b503d58310b523553c-Abstract-Conference.html.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. FLUX.1 kontext: Flow matching for in-context image generation and editing in latent space. *CoRR*, abs/2506.15742, 2025. doi: 10.48550/ARXIV.2506.15742. URL https://doi.org/10.48550/arXiv.2506.15742.

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.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online RL. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025a. URL http://papers.nips.cc/paper_files/paper/2025/hash/3a10c46572628d58cb44fb705f25cbbf-Abstract-Conference.html.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, Guopeng Li, Yuang Peng, Quan Sun, Jingwei Wu, Yan Cai, Zheng Ge, Ranchen Ming, Lei Xia, Xianfang Zeng, Yibo Zhu, Binxing Jiao, Xiangyu Zhang, Gang Yu, and Daxin Jiang. Step1x-edit: A practical framework for general image editing. *CoRR*, abs/2504.17761, 2025b. doi: 10.48550/ARXIV.2504.17761. URL https://doi.org/10.48550/arXiv.2504.17761.

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.

Yancheng Long, Yankai Yang, Hongyang Wei, Wei Chen, Tianke Zhang, Haonan Fan, Changyi Liu, Kaiyu Jiang, Jiankang Chen, Kaiyu Tang, Bin Wen, Fan Yang, Tingting Gao, Han Li, and Shuo Yang. Spatialreward: Bridging the perception gap in online RL for image editing via explicit spatial reasoning. *CoRR*, abs/2602.07458, 2026. doi: 10.48550/ARXIV.2602.07458. URL https://doi.org/10.48550/arXiv.2602.07458.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *CoRR*, abs/2310.04378, 2023. doi: 10.48550/ARXIV.2310.04378. URL https://doi.org/10.48550/arXiv.2310.04378.

Xin Luo, Jiahao Wang, Chenyuan Wu, Shitao Xiao, Xiyan Jiang, Defu Lian, Jiajun Zhang, Dong Li, and Zheng Liu. Editscore: Unlocking online RL for image editing via high-fidelity reward modeling. *CoRR*, abs/2509.23909, 2025. doi: 10.48550/ARXIV.2509.23909. URL https://doi.org/10.48550/arXiv.2509.23909.

Yihong Luo, Tianyang Hu, Weijian Luo, and Jing Tang. TDM-R1: reinforcing few-step diffusion models with non-differentiable reward. *CoRR*, abs/2603.07700, 2026. doi: 10.48550/ARXIV.2603.07700. URL https://doi.org/10.48550/arXiv.2603.07700.

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.

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. 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=di52zR8xgf.

Yusu Qian, Jiasen Lu, Tsu-Jui Fu, Xinze Wang, Chen Chen, Yinfei Yang, Wenze Hu, and Zhe Gan. Gie-bench: Towards grounded evaluation for text-guided image editing. *CoRR*, abs/2505.11493, 2025. doi: 10.48550/ARXIV.2505.11493. URL https://doi.org/10.48550/arXiv.2505.11493.

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. URL https://doi.org/10.1109/CVPR52688.2022.01042.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. 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=TIdIXIpzhoI.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300. URL https://doi.org/10.48550/arXiv.2402.03300.

Jiayi Song, Shijie Huang, Fangtai Wu, Yubo Huang, Zhenxiong Tan, Songhua Liu, Jiaming Liu, and Ru Huang. EDITBRIDGE: towards faithful and efficient ultra-high-resolution image editing. *CoRR*, abs/2608.18063, 2026. doi: 10.48550/ARXIV.2608.18063. URL https://doi.org/10.48550/arXiv.2608.18063.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Fu-Yun Wang, Han Zhang, Michaël Gharbi, Hongsheng Li, and Taesung Park. Promptrl: Prompt matters in RL for flow-based image generation. *CoRR*, abs/2602.01382, 2026a. doi: 10.48550/ARXIV.2602.01382. URL https://doi.org/10.48550/arXiv.2602.01382.

Haozhe Wang, Cong Wei, Weiming Ren, Jiaming Liu, Fangzhen Lin, and Wenhu Chen. Rationalrewards: Reasoning rewards scale visual generation both training and test time. *CoRR*, abs/2604.11626, 2026b. doi: 10.48550/ARXIV.2604.11626. URL https://doi.org/10.48550/arXiv.2604.11626.

Su Wang, Chitwan Saharia, Ceslee Montgomery, Jordi Pont-Tuset, Shai Noy, Stefano Pellegrini, Yasumasa Onoe, Sarah Laszlo, David J. Fleet, Radu Soricut, Jason Baldridge, Mohammad Norouzi, Peter Anderson, and William Chan. Imagen editor and editbench: Advancing and evaluating text-guided image inpainting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 18359–18369. IEEE, 2023. doi: 10.1109/CVPR52729.2023.01761. URL https://doi.org/10.1109/CVPR52729.2023.01761.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *CoRR*, abs/2306.09341, 2023. doi: 10.48550/ARXIV.2306.09341. URL https://doi.org/10.48550/arXiv.2306.09341.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/33646ef0ed554145eab65f6250fab0c9-Abstract-Conference.html.

Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. Imgedit: A unified image editing dataset and benchmark. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/bec783f119b969772c35243b1c801090-Abstract-Datasets_and_Benchmarks_Track.html.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024a. URL http://papers.nips.cc/paper_files/paper/2024/hash/54dcf25318f9de5a7a01f0a4125c541e-Abstract-Conference.html.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 6613–6623. IEEE, 2024b. doi: 10.1109/CVPR52733.2024.00632. URL https://doi.org/10.1109/CVPR52733.2024.00632.

A METHOD DETAILS

This appendix develops the DMD background and SATP analysis, details how PRA-RL augments exploration and updates the editor, and describes joint training. It concludes with the rewriter's training and evaluation.

A.1 DMD FOUNDATIONS AND FAKE-SCORE TRAINING

Score parameterization. For $\alpha_t = 1 - \sigma_t$ and $\sigma_t > 0$, clean denoising estimates give

$$\widehat{s}_T = \frac{\alpha_t T_t - x_t}{\sigma_t^2}, \qquad \widehat{s}_\phi = \frac{\alpha_t F_t - x_t}{\sigma_t^2}, \qquad \widehat{s}_\phi - \widehat{s}_T = \frac{\alpha_t}{\sigma_t^2}(F_t - T_t). \tag{16}$$

Distribution-matching gradient. We derive the distribution-matching gradient in Section 3. Fix the condition $h = (S, c_0)$ and let $x_\theta = G_\theta(z; h)$, where $z \sim \mathcal{N}(0, I)$ contains the generator's sampling randomness. Let $q_{\theta,t}$ be the distribution of $x_t = \alpha_t x_\theta + \sigma_t \epsilon$, with $\alpha_t = 1 - \sigma_t$ and $\epsilon \sim \mathcal{N}(0, I)$. For an idealized teacher distribution $p_{T,t}$ and positive time weight $\omega(t)$, consider

$$J_{\mathrm{DM}}(\theta) = \mathbb{E}_t\big[\omega(t) D_{\mathrm{KL}}\big(q_{\theta,t}(\cdot \mid h) \| p_{T,t}(\cdot \mid h)\big)\big]. \tag{17}$$

Assume smooth positive densities at the sampled noise levels, valid interchange of differentiation and integration, and vanishing boundary terms. Write $s_{q_\theta} = \nabla_{x_t} \log q_{\theta,t}$, $s_T = \nabla_{x_t} \log p_{T,t}$, and $J_\theta x_\theta = \partial x_\theta / \partial \theta$. Differentiating the reparameterized expectation gives the score-difference form used in DMD (Yin et al., 2024b;a):

$$\begin{aligned} \nabla_\theta J_{\mathrm{DM}} &= \mathbb{E}\big[\omega(t)\alpha_t (J_\theta x_\theta)^\mathsf{T}(s_{q_\theta} - s_T)\big] \\ &\approx \mathbb{E}\left[\omega(t)\frac{\alpha_t^2}{\sigma_t^2}(J_\theta x_\theta)^\mathsf{T}(F_t - T_t)\right]. \end{aligned} \tag{18}$$

The explicit density-parameter term has zero expectation because $\int q_{\theta,t}\, \partial_\theta \log q_{\theta,t}\, dx_t = \partial_\theta \int q_{\theta,t}\, dx_t = 0$. The second line substitutes the denoiser-based score estimates from Equation 16; it becomes exact for exact denoisers. One factor of α_t comes from re-noising the clean prediction, and the other comes from the score–denoiser correspondence.

This derivation uses the Jacobian of the full generator and explains the origin and sign of $F_t - T_t$. Our training computes a local update by detaching the cached state and differentiating the recomputed final prediction. It uses a learned fake estimator, a teacher with classifier-free guidance (Ho & Salimans, 2022), and sample-dependent normalization. Equation 24 gives this local surrogate gradient after source anchoring.

Fake-score training. During joint refinement, let x denote a detached clean prediction from a cached original-instruction rollout. The fake estimator minimizes

$$\mathcal{L}_{\mathrm{fake}} = \mathbb{E}_{x,t,\epsilon}\frac{w(t)}{D}\left\| v_\phi((1-\sigma_t)x + \sigma_t\epsilon, \sigma_t; S, c_0) - (\epsilon - x)\right\|_2^2. \tag{19}$$

Here $w(t)$ weights the sampled noise level. This loss fits the estimator to original-instruction student outputs with a separate optimizer and supplies no gradient to the editor. The distillation configuration uses uniform fake-training indices from 0 to 999 with $w(t) = 1$, and a logit-normal SATP re-noising sampler restricted to indices 20–980. Teacher guidance is 4 and fake guidance is 1.

A.2 SOURCE-ANCHORED TARGET PROJECTION

Projection identity. Vectorize the latent into D coordinates. For a teacher prediction T and its selected binary mask M, let $\mathcal{P}$ be the anchored coordinates and $\mathcal{U}$ their complement. The constraint $M \odot X = M \odot S$ fixes $X_\mathcal{P} = S_\mathcal{P}$. For every feasible X,

$$\|X - T\|_2^2 = \|X_\mathcal{U} - T_\mathcal{U}\|_2^2 + \|S_\mathcal{P} - T_\mathcal{P}\|_2^2. \tag{20}$$

The second term is fixed, and the first is uniquely minimized by $X_\mathcal{U} = T_\mathcal{U}$. Hence the unique minimizer is $T^{\mathrm{SA}} = (T_\mathcal{U}, S_\mathcal{P})$, proving Equation 5. This result holds for each evaluated mask, including masks chosen adaptively from T and S.

Effect on the denoising target. Let Y denote an ideal edited latent. To quantify the effect of source anchoring, we compare the teacher target and its projection against Y. Separating the anchored and unanchored coordinates gives

$$\begin{aligned} \|T^{\mathrm{SA}} - Y\|_2^2 &= \|(1-M)\odot(T-Y)\|_2^2 + \|M\odot(S-Y)\|_2^2, \\ \|T - Y\|_2^2 &= \|(1-M)\odot(T-Y)\|_2^2 + \|M\odot(T-Y)\|_2^2. \end{aligned} \tag{21}$$

Subtracting yields the exact error change:

$$\|T^{\mathrm{SA}} - Y\|_2^2 - \|T - Y\|_2^2 = \|M\odot(S-Y)\|_2^2 - \|M\odot(T-Y)\|_2^2. \tag{22}$$

Equation 22 shows that anchoring reduces target error whenever the source is closer to Y than the teacher on the selected coordinates, measured by their total squared error. The target error on unanchored coordinates remains unchanged.

Effect of normalization. Let $n_{\mathrm{DMD}} = D^{-1}\|x_\theta - T_t\|_1$ and $\bar{g}_{\mathrm{DMD}} = (F_t - T_t)/n_{\mathrm{DMD}}$. For finite positive normalizers,

$$\bar{g}_{\mathrm{SA}} - \bar{g}_{\mathrm{DMD}} = \frac{M_t \odot (T_t - S)}{n_{\mathrm{SA}}} + \left(\frac{1}{n_{\mathrm{SA}}} - \frac{1}{n_{\mathrm{DMD}}}\right)(F_t - T_t). \tag{23}$$

The first term is the anchor correction, while the second accounts for the change in sample-wide scaling. The unanchored coordinates retain the same unnormalized signal, with its magnitude rescaled by the new normalizer.

Numerical implementation and local gradient. The source latent is $S = \mathcal{E}(I_s)$, where the frozen encoder $\mathcal{E}$ includes the model's latent normalization. The frozen decoder $\mathcal{D}$ maps output latents back to images. SATP compares and combines source and predicted latents in this shared normalized domain. The implementation computes agreement in float32, detaches the mask and target, and applies nonfinite-value handling after gradient normalization. The mask in Equation 3 uses $\gamma = 3$, $\tau = 0.5$, and $\varepsilon_m = 10^{-8}$. Equation 13 describes finite positive normalizers. After division, the implementation applies `torch.nan_to_num` to the normalized signal. Target replacement uses the binary mask; the continuous agreement values are used for visualization.

For a cached input state and a finite positive normalizer, differentiating the stop-gradient surrogate gives

$$\nabla_\theta \mathcal{L}_{\mathrm{SATP}} = \frac{1}{D}(J_\theta x_\theta)^{\mathsf{T}} \frac{F_t - T_t^{\mathrm{SA}}}{n_{\mathrm{SA}}}. \tag{24}$$

The mask, teacher and fake predictions, and normalizer are evaluated before this differentiation and held fixed. The Jacobian is that of the recomputed final prediction, with its cached input state detached.

The original-step anchoring gate is $\sigma_{K-1} < 1$; it is not a threshold on the separately sampled σ_t.

A.3 Prompt-Rewrite-Augmented Reinforcement Learning

PRA-RL augments stochastic rollout exploration with alternative source-conditioned instructions. Noise varies the sampled actions, while rewrites provide additional conditioning choices for the editor. The original request remains the reward objective for every rollout.

Instruction and noise allocation. Each task uses the original instruction and three rewrites, with four noise repetitions per instruction. Rewrites act only at the selected stochastic transition. Initial noise is shared across the group; at the selected transition, each repetition shares its noise across candidate instructions, while all other steps share noise across the entire group. This forms the paired instruction–noise grid in Section 3.2.

Original-task rewards and group normalization. We use the frozen MoRE-OPD model (Anonymous, 2027) as the RL reward model. It evaluates each final edit against the source image and original instruction, rather than the rewritten instruction.

Each original source–instruction task defines one group with $J = 4$ instruction candidates and $N = 4$ noise repetitions. All 16 terminal rewards enter the same group mean and population standard

deviation, independently of other tasks. We use $\epsilon_A = 10^{-8}$, group-local standard deviations, and no advantage clipping. The clipping in Equation 15 applies to the policy ratio. Original tasks remain separate groups even if their instruction strings coincide.

The original-instruction row is retained for SATP and fake-model fitting. Candidate instructions and paired noise determine rollout collection; the policy update uses the standard clipped GRPO loss on the selected transition. In particular, group normalization covers all candidates for the original task rather than normalizing each rewrite separately.

Transition parameterization. Let v_θ be the velocity network underlying G_θ. At step k, its clean prediction is

$$x_\theta^{(k)} = y_k - \sigma_k v_\theta(y_k, \sigma_k; S, c^{(k)}). \tag{25}$$

The sampling step k is distinct from the DMD re-noising time t. In $G_\theta(z; h)$, z collects the initial latent noise and all independent Gaussian transition noises. The output is $y_K = x_\theta$. The final deterministic step gives $x_\theta = x_\theta^{(K-1)}$.

Following the SDE policy formulation of Flow-GRPO (Liu et al., 2025a), we use Gaussian transitions with a constant diffusion coefficient η at stochastic steps. Writing $d_k = \sigma_k - \sigma_{k+1} > 0$, the transition is

$$y_{k+1} = \mu_{\theta,k}(y_k; S, c) + \nu_k \xi_k, \qquad \xi_k \sim \mathcal{N}(0, I), \tag{26}$$

with

$$\begin{aligned} \mu_{\theta,k}(y_k; S, c) &= \left(1 - \frac{\eta^2 d_k}{2\sigma_k}\right) y_k - d_k \left(1 + \frac{\eta^2(1-\sigma_k)}{2\sigma_k}\right) v_\theta(y_k, \sigma_k; S, c), \\ \nu_k &= \eta\sqrt{d_k}. \end{aligned} \tag{27}$$

The final transition uses $\eta = 0$ and $d_{K-1} = \sigma_{K-1}$, recovering the clean prediction in Equation 25.

Log-probability and clipping. We average the elementwise Gaussian log-probabilities across latent coordinates. Writing $\ell_{\theta,j,i}$ for this average at the selected stochastic transition gives

$$\ell_{\theta,j,i} = -\frac{\|y_{j,i,k^\star+1} - \mu_{\theta,k^\star}(y_{j,i,k^\star}; S, c_j)\|_2^2}{2D\nu_{k^\star}^2} - \frac{1}{2}\log(2\pi\nu_{k^\star}^2). \tag{28}$$

The ratio used in Equation 15 is

$$\rho_{j,i}(\theta) = \exp\big(\ell_{\theta,j,i} - \mathrm{sg}[\ell_{\theta_{\mathrm{old}},j,i}]\big). \tag{29}$$

It is equivalently the Dth root of the ratio of the joint Gaussian densities. Both policies evaluate the same cached state, sampled next state, candidate instruction, and transition variance. Old-policy log-probabilities are recomputed once under the collection policy before optimization and then held fixed across update passes. The clipping range and update settings are given in Appendix A.4.

Gradient through one transition. The full rollout and reward evaluation run without gradients. For the GRPO update, PRA-RL restricts the replay indices to $\{k^\star\}$ and supplies the instruction c_j used at that step. Both cached states $y_{j,i,k^\star}$ and $y_{j,i,k^\star+1}$ are detached. Consequently,

$$\nabla_\theta \ell_{\theta,j,i} = \left(\frac{\partial \mu_{\theta,k^\star}}{\partial \theta}\right)^{\mathsf{T}} \frac{y_{j,i,k^\star+1} - \mu_{\theta,k^\star}}{D\nu_{k^\star}^2}. \tag{30}$$

Differentiating Equation 15 scales or clips this local policy gradient using the frozen advantage and old-policy anchor. All other rollout transitions contribute to the terminal reward but supply no GRPO gradient. The parameters are shared across denoising steps, so this selected-step update trains the same generator used throughout the trajectory. The SATP gradient is computed separately from the original-instruction final prediction.

A.4 Joint Training

Training configuration. Training starts with the cold-start distillation procedure of RTDMD (Huang et al., 2026) under the original editing instruction. Cold start uses full-parameter training for 15,000 steps with learning rate 10^{-5}. The resulting student and fake-score checkpoints initialize

joint SATP and PRA-RL refinement. In this second stage, the editor is trained with LoRA rank 16 and alpha 32, learning rate 5×10^{-5}, and approximately 1,000 steps on eight H20 servers. We set λ_D=0.01 and λ_R=1. The 1K training set contains approximately 240,000 high-quality internal editing examples; the 4K set contains several hundred thousand high-quality internal examples. These are the few-step training data, distinct from the base-editor preparation in Appendix B.1 and the rewriter's SFT data. The native 1K implementation uses the Klein autoencoder and official noise shift (Labs, 2025). The policy update uses diffusion coefficient $\eta = 0.7$, clipping range $\delta = 0.005$ held constant throughout training, two update passes per batch, and no reference-KL penalty. PRA-RL defines the few-step instruction and noise allocation described in Section 3.2.

Update sequence. In Stage II, starting from the student and fake estimator initialized by the RT-DMD cold-start stage, we use $K = 4$ sampling steps. GRPO updates the selected stochastic transition $k^\star \in \{0, 1, 2\}$, while SATP updates the original-instruction final prediction at step 3 through the same editor parameters. Each training iteration proceeds as follows:

1. Sample source–instruction tasks and obtain the original plus three rewritten instructions for each task. Select one stochastic step uniformly for each group, shared across its candidates.
2. Collect the 4×4 candidate grid per task using Equations 7 and 9; cache states, sampled transitions, conditions, and original-prompt trajectories.
3. Evaluate all final outputs against their original requests and construct an advantage for each rollout (j, i) using Equation 14.
4. Replay the selected transition once under the collection policy to cache its old log-probability, before updating the generator. Keep these log-probabilities and the advantages fixed across both passes.
5. Fit the fake estimator on detached original-prompt clean predictions, with each original rollout included once in the fake-training sample set.
6. In each of the two student update passes, recompute the original-prompt final prediction for SATP. Re-noise it, evaluate teacher and fake without gradients, and construct M_t and T_t^{SA}. Also replay the selected stochastic transition under its candidate instruction for the clipped GRPO loss. Accumulate the two weighted losses' gradients according to Equation 11, then update the student adapters after gradient clipping. Reuse the cached rollout states, advantages, and old log-probabilities across both passes.

The teacher, source encoder, rewriter, and evaluator parameters are frozen throughout refinement. The student and fake-score adapters have separate optimizers. Neither target replacement nor prompt exploration adds a network evaluation to the deployed editor.

A.5 Instruction Rewriter

Training data and objective. Our instruction rewriter uses a Qwen3-VL-9B conditional-generation model. We construct instruction-expansion supervision from open editing datasets, using a stronger model to produce instruction pairs. Each example contains a source image, a concise request, and a detailed target instruction. The rewriter learns to generate the detailed instruction from the image and concise request; the associated edited image is retained for provenance.

Supervised fine-tuning. We fine-tune the rewriter on 13,270 instruction-expansion examples using LoRA with rank 64. The adapter is merged into the model for inference, and the rewriter remains frozen throughout editor training.

For each task, the frozen rewriter provides three candidates alongside its original instruction. The candidates remain fixed during rollout collection and policy updates.

Qualitative examples. Figure 4 illustrates how expanded instructions specify lighting, pose, and target appearance while making preservation requirements explicit. The resulting edits better retain the mountain structure and reading pose, and more closely follow the requested standing posture in these examples.



Figure 4: **Editing with original and rewritten instructions.** Each example shows the source image and the two editing results, together with their instructions.

Inference-time effectiveness. Table 6 compares original-instruction and rewritten-instruction inference using the trained VINS-Distill editors at 4K and 1K. The editor is fixed within each comparison, and the rewritten instruction is used throughout four-step sampling. Evaluation refers to the original source–instruction request. The original-instruction rows reproduce the corresponding main results.

Table 6: **Inference-time rewriting with a fixed editor.** Bold marks the better result. Gain is rewritten minus original; SR gains are percentage points.

	MoRE-OPD (Anonymous, 2027)					Additional Reward	
Instruction	IF ↑	NC ↑	VQ ↑	Score ↑	SR (%) ↑	SpatialReward ↑ (Long et al., 2026)	RationalRewards ↑ (Wang et al., 2026b)
VINS-120K++							
Original	8.5386	9.0506	8.7802	8.4831	60.79	7.6784	0.8071
Rewritten	**8.8751**	**9.1208**	**8.8611**	**8.5225**	**63.48**	**7.6978**	**0.8086**
Gain	+0.3365	+0.0702	+0.0809	+0.0394	+2.69	+0.0194	+0.0015
GEdit-Bench							
Original	8.4257	8.7624	**8.9026**	8.1997	55.28	7.4130	0.7773
Rewritten	**8.7013**	**8.7673**	8.8168	**8.3596**	**56.44**	**7.4151**	**0.7792**
Gain	+0.2756	+0.0049	−0.0858	+0.1599	+1.16	+0.0021	+0.0019
ImgEdit-Bench							
Original	7.8653	8.9898	9.0114	7.8562	53.37	7.0130	0.7874
Rewritten	**7.8672**	**9.0125**	**9.0251**	**7.9256**	**55.78**	**7.0249**	**0.7927**
Gain	+0.0019	+0.0227	+0.0137	+0.0694	+2.41	+0.0119	+0.0053

Rewriting improves overall Score and success rate on all three benchmarks. On VINS-4K-Bench, it improves every reported metric, including instruction following and source preservation. On ImgEdit-Bench, improvements are also consistent across metrics. On GEdit-Bench, better instruction following and overall Score accompany a small reduction in visual quality. These results show that the rewriter supplies useful alternative editing conditions for the student. The training-time value of this conditioning diversity is evaluated separately by the prompt-conditioning ablation in Table 5. The main VINS-Distill comparisons use original-instruction inference.

System prompt. The rewriter receives the source image and editing request with the following system prompt:

```
You will act as a top-tier Image Editing AI. Your core task
is to deeply analyze the user's text input and the reference
image context to transform it into a detailed, artistic, and
fully user-intent-compliant image edit.
Your workflow is divided into two phases:
1. Thinking Phase (<think>): In the <think> tag, you need
to conduct a structured thinking process. You must adopt a
task-diagnostic approach:
1. Diagnose Task: Identify the edit type and analyze key
requirements.
2. Prioritize Analysis:
* Adding: Analyze the new element's position and
appearance, ensuring seamless integration with the original
image's lighting, shadows, and style.
* Removing: Identify the target for removal and determine
how to logically fill the resulting space using surrounding
textures and lighting.
* Modifying: Analyze what to change and what it should
become, while emphasizing which elements must remain
unchanged.
```

```
* Style Transfer: Deconstruct the target style into
specific features (e.g., brushstrokes, color palette) and
apply them to the original image.
* Text Editing: Ensure correct content and format.
Consider the text's visual style (e.g., font, color,
material) and how it adapts to the surface's perspective,
curvature, and lighting.
* Reference Editing: Extract specific visual elements
(e.g., appearance, posture, composition, lines, depth) from
the reference image to generate an image that aligns with
the text description while also incorporating the referenced
content.
* Inferential Editing: Identify vague requests (e.g., "make
it more professional") and translate them into concrete
visual descriptions.
2. Recaption Phase (<recaption>): In the <recaption> tag,
merge all the key details from the thinking process into a
coherent, precise, and visually evocative final description.
You must adhere to the following principles:
* Faithful to Intent: All edits must be highly consistent
with the user's core intent.
* Style Handling: If the user specifies a style, strictly
follow it. Otherwise, maintain the original image's style.
* Text Rendering: Text to be rendered must be enclosed in
double quotes ("").
* Preserve Unchanged Elements: Emphasize elements that
**remain unchanged**. Unless explicitly instructed, never
alter a character's identity/appearance, the core
background, camera angle, or overall style.
* Clear Editing Instructions:
* Replacement: Use the logic "**replace B with A**," and
provide a detailed description of A.
* Addition: Clearly state what to add, where, and what it
looks like.
* Unambiguous Referencing: Avoid vague references (e.g.,
"that person"). Use specific descriptions of appearance.
* Absolutely Objective: Describe only what is visually
present. Avoid subjective words.
* Physical and Logical Consistency: All scene elements must
strictly adhere to physical laws.
* Structured Description: Follow a logical order
(background to foreground).
* Use Present Tense: Describe from an observer's
perspective.
* Language Requirement: Ensure all output is strictly in
English, except for specific text elements that need to be
rendered.
```

B Experimental Details

We describe the base editor, benchmark, and evaluation protocol, followed by ablation details, inference efficiency, and additional qualitative comparisons.

B.1 4K Base Editor and Training Data

We prepare the 50-step 4K base editor from FLUX.2-klein-base-9B by adapting it to a more compressed latent representation and then fine-tuning on high-quality editing data. The autoencoder's spatial downsampling factor is increased progressively from 8 to 32 and 64, with the editor's input and output projections adapted at each stage. Quality fine-tuning proceeds from 1024×1024 to

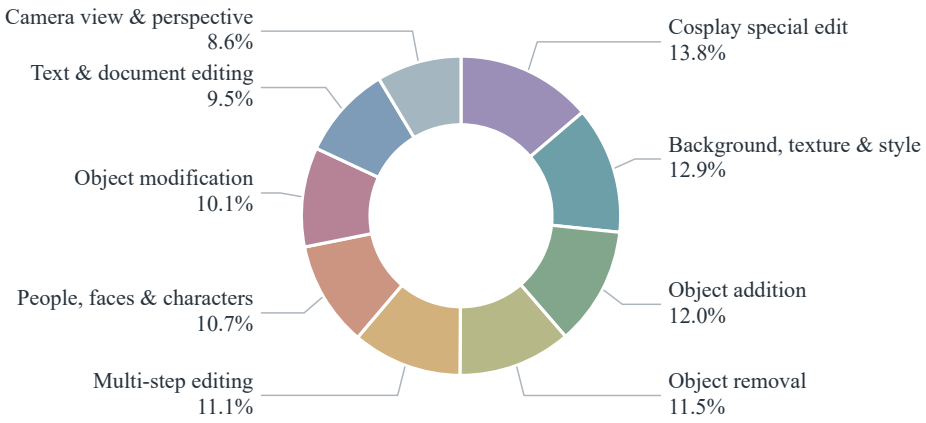

Figure 5: **Editing-task distribution of VINS-4K-Bench.** The benchmark contains 3,305 tasks across nine categories. Percentages are computed from task counts and rounded to one decimal place.

native 4096×4096, preserving image aspect ratios and aligning dimensions to the autoencoder's downsampling factor. This preparation provides the teacher for the few-step training in Section 4.

The data pipeline combines public editing data with internally annotated pairs. Its 4K data includes video frame pairs that form plausible editing tasks, with corresponding instructions produced by a rewriter, and data constructed through super-resolution of selected high-quality 1K images. A multimodal quality filter assesses visual quality, aesthetics, information density, and edit accuracy. MoRE-OPD further scores each source–edit–instruction triple for editing quality. The training data covers spatial, text, portrait, object, background and style, and multi-step edits. The evaluation set below separately characterizes the tasks used to assess the trained editor.

B.2 VINS-4K-Bench Composition

Construction and evaluation tasks. VINS-4K-Bench expands the 505-sample evaluation set of VINS-120K (Chen et al., 2026) to 3,305 editing tasks. We broaden the image-scene categories and manually curate instructions to cover diverse editing demands. All evaluation images are native 4K. They are drawn from high-quality internal 4K collections and publicly available high-quality online data. Each task pairs a source image with an English editing instruction and includes a task category and fine-grained subtype. Evaluation uses the original instruction, consistent with the main comparisons in Section 5.1.

Editing coverage. Figure 5 shows the distribution across the nine task categories; Table 7 gives exact counts. Their shares range from 8.6% to 13.8%, and the dataset contains 187 distinct subtype labels. Object addition, removal, and modification test localized changes and preservation of the surrounding content. Background, style, and camera-view edits involve broader transformations. Text editing tests precise content rendering, while compound instructions require multiple requested changes within one task. The cosplay-specific category extends coverage to character and costume-related edits. This mixture evaluates preservation in the context of different editing requirements.

Source-image content. Table 8 groups tasks by the source-image directory labels. These are content categories, distinct from the editing operations above: for example, a cosplay image can be paired with an ordinary object or background edit. Cosplay, landscapes, portraits, scene text, plants and flowers, and food together account for 91.8% of the tasks. They provide varied textures, facial and object structures, and lettering for evaluating source fidelity. The remaining categories add animals, buildings, transportation, objects, and other scenes. Counts are per editing task, rather than counts of unique source images.

Table 7: **Editing tasks in the 4K benchmark.** The 3,305 tasks cover nine categories and 187 distinct subtype labels. Percentages are rounded independently.

Task category	Tasks	Share (%)
Cosplay special edit	455	13.8
Alter background, texture, or style	426	12.9
Add object or element	396	12.0
Remove object or element	379	11.5
Multi-step instruction (compound edits)	367	11.1
Edit people, faces, or character attributes	352	10.7
Modify existing object	333	10.1
Edit text or document content	313	9.5
Change camera view or perspective	284	8.6
Total	3,305	100.0

Table 8: **Source-image content in the 4K benchmark.** Categories follow source-directory labels; counts refer to editing tasks. Percentages are rounded independently.

Source-image category	Tasks	Share (%)
Cosplay	779	23.6
Landscape	717	21.7
Portrait	537	16.2
Text in scene	432	13.1
Plants and flowers	329	10.0
Food	241	7.3
Animals	44	1.3
Objects	41	1.2
Human	35	1.1
Building	33	1.0
Transportation	32	1.0
Scene	30	0.9
Compose	30	0.9
Text	25	0.8
Total	3,305	100.0

B.3 Evaluation Protocol

Image-editing evaluation considers both the requested modification and the preservation of other content (Wang et al., 2023; Basu et al., 2023; Qian et al., 2025). Whereas ImageReward, PickScore, and HPSv2 model preferences for generated images (Xu et al., 2023; Kirstain et al., 2023; Wu et al., 2023), editing-oriented evaluators use the source and instruction to assess the requested transformation (Luo et al., 2025; Long et al., 2026; Wang et al., 2026b). Our evaluation accordingly reports instruction following, preservation, and visual quality, alongside scores from additional reward models.

Training settings are summarized in Appendix A.4. All main quality comparisons use four-step students and evaluate outputs against the original source–instruction task.

MoRE-OPD reports IF, NC, VQ, Score, and success rate on each benchmark. SpatialReward and RationalRewards are reported separately from these MoRE-OPD outputs. We use the same evaluator settings across methods within each comparison. We follow the released GEdit-Bench evaluation protocol (Liu et al., 2025b). On GEdit-Bench, the success counts are 312/605 for Klein-9B and 335/606 for VINS-Distill. On ImgEdit-Bench, VINS-Distill succeeds on 420/787 samples, excluding multi-turn tasks. For VINS-4K-Bench, the counts are 1,675/3,305 for RTDMD and 2,009/3,305 for VINS-Distill.

B.4 Ablation Details and Additional Results

Shared configuration. The three ablation tables contain seven distinct refinement configurations and a cold-start reference. Refinement runs share the cold-start checkpoint, training data, number of updates, and four-step sampler. In runs with distillation, the distillation loss and fake-model fitting use four original-instruction outputs per task.

Framework variants. Cold Start evaluates the checkpoint before refinement. The *w/ SATP* variant optimizes only the distillation term in Equation 11, setting $\lambda_R = 0$, and maintains the auxiliary fake-model update. The *w/ PRA-RL* variant optimizes only the GRPO term, setting $\lambda_D = 0$, with the same prompt–noise grouping and selected-transition update as PRA-RL. It requires no teacher or fake-model training during refinement. The joint variant retains both terms and their full-method weights. The active loss in each single-branch variant uses the same weight as in the joint model. All three runs start independently from the cold-start checkpoint.

Figure 6 complements the framework ablation in Table 3, showing the requested lens-flare removal and preservation of the surrounding scene across these variants.



Figure 6: **Qualitative framework ablation.** Cold Start, SATP alone, PRA-RL alone, and their joint training are compared on the same editing task.

Distillation variants. Ordinary DMD uses the unmodified teacher target T_t, including in the normalizer of Equation 13. The direct source-supervision control adds the following term to ordinary DMD:

$$\mathcal{L}_{\text{src}} = \frac{\lambda_{\text{src}}}{2D} \left\| \text{sg}[M_t] \odot (x_\theta - S) \right\|_2^2 , \tag{31}$$

where M_t is computed with the same teacher–source agreement rule and thresholds as SATP. The mask is detached, and the source term differentiates the final student prediction used for distillation. All three distillation variants retain the same PRA-RL configuration.

Prompt-conditioning variants. This comparison evaluates noise-only exploration against the additional conditioning diversity supplied by rewrites, and tests whether rewrites should act locally or throughout the trajectory. Without prompt rewriting, all 16 rollouts use c_0. They share the initial noise and the noise at non-selected steps, while using 16 independent noise draws at $k^\star$. Four rollouts, selected before reward evaluation, supply the distillation and fake-model samples. This retains the rollout budget without duplicating identical original-instruction trajectories.

Full-trajectory prompt rewriting retains the $J = N = 4$ group, candidate rewrites, and noise-pairing rule of PRA-RL. It uses c_j at every step of candidate j; the $j = 0$ row continues to use c_0 throughout. Only the selected transition receives GRPO gradients, as in the full method. Rewards are evaluated against the original task and normalized over all 16 outputs. Both prompt-conditioning controls retain SATP. All trained variants use the original instruction at inference.

Training reward trajectories. Figure 7 complements the final evaluation in Table 5. All three variants improve early in training. The full SATP + PRA-RL model attains higher training rewards toward the end of refinement, while the two prompt-conditioning controls remain at similar levels.



Figure 7: **Training rewards for the prompt-conditioning ablation.** With SATP retained, we compare training without prompt rewriting, full-trajectory rewriting, and PRA-RL.

Table 9: **4K editing inference time on a P800 XPU.** Times are approximate per-image measurements. Speedup is relative to the teacher under the listed sampling configurations.

Model	Steps	CFG	Time (s) ↓	Speedup ↑
4K teacher	50	Scale 4	≈ 170	$1\times$
VINS-Distill	4	None	≈ 12	$\approx 14\times$

B.5 Inference Efficiency

Table 9 reports approximate per-image times on a P800 XPU at 4K resolution. The 50-step teacher with CFG scale 4 takes approximately 170 seconds, while the four-step distilled editor without CFG takes approximately 12 seconds, giving about $14\times$ faster inference. This comparison reflects both fewer denoising steps and removal of CFG.

B.6 Qualitative Comparisons at 1K

Figure 8 compares the requested edits and preservation of the surrounding scene across the 1K models.

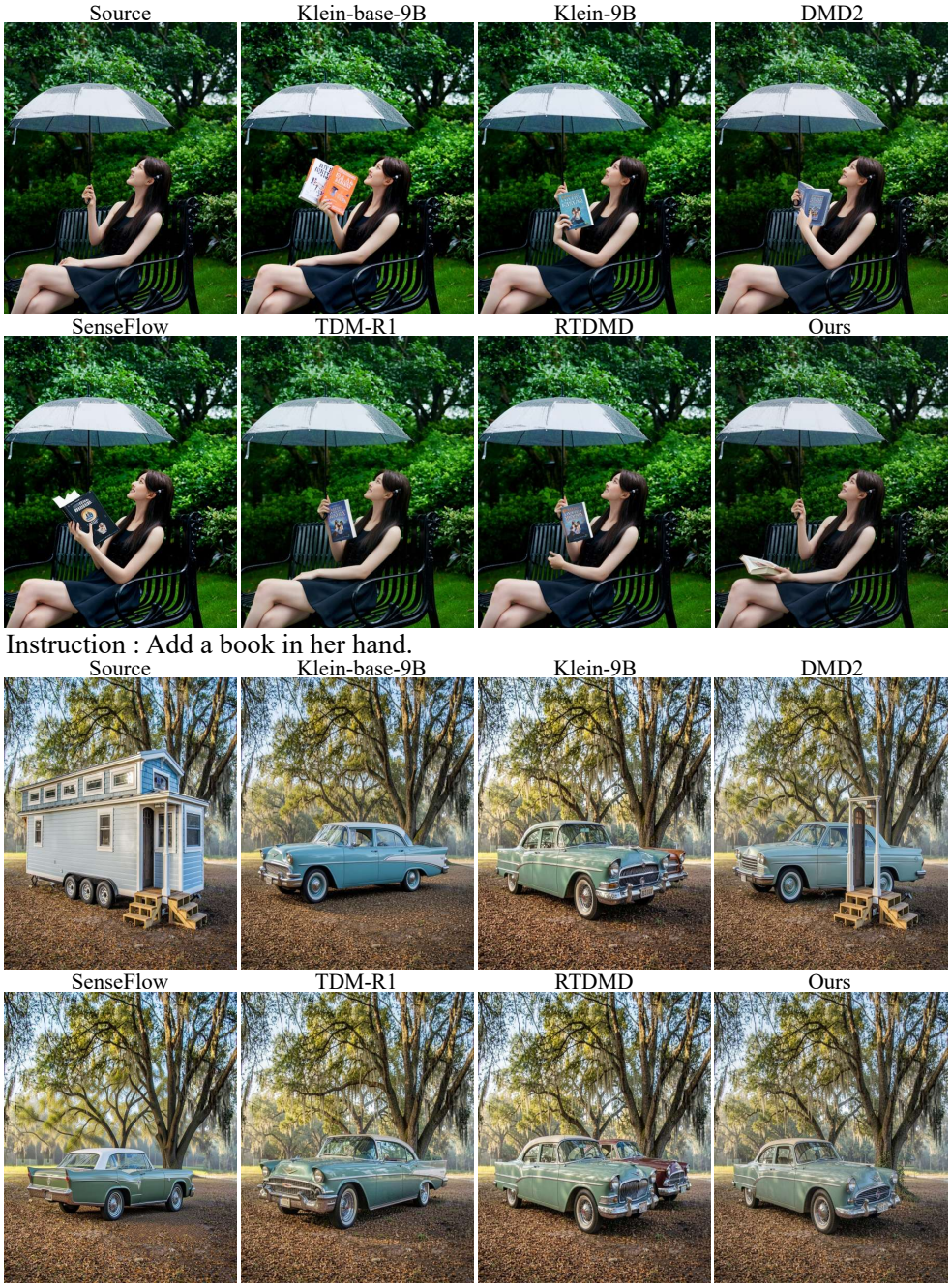

Figure 8: **Qualitative comparison at 1K.** Comparisons show editing results and preservation of the surrounding scene.