SPURFLOW: SPATIALLY-ADAPTIVE RESIDUAL STEERING FOR FEW-STEP IMAGE EDITING

Anonymous authors
Paper under double-blind review

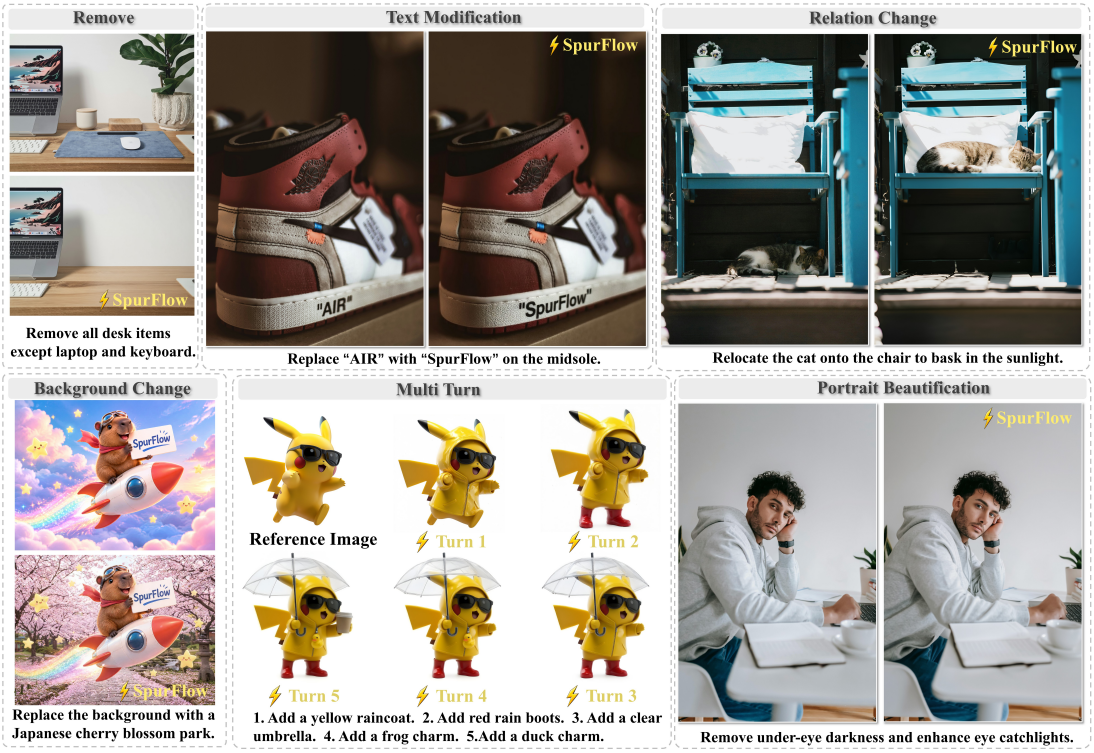

Figure 1: Qualitative results of 2-step SPURFLOW distilled from Qwen-Image-Edit-2511 (Qwen Team, 2025), covering text modification, relation changes, multi-turn editing, and beautification.

ABSTRACT

Recent diffusion distillation methods enable high-quality image generation with only a few sampling steps, yet extending them to image editing remains challenging. We argue that this difficulty arises from a mismatch between conventional distillation objectives and the asymmetric nature of image editing. Unlike generation, most editing tasks preserve the majority of the reference image while modifying only sparse instruction-specified regions. Under full-field matching, preserved tokens dominate both spatially and in gradient contribution, diluting sparse yet critical edit signals. To address this, we propose SPURFLOW, an editing-oriented distillation framework that separates reference-preserving content from edit-specific updates. SPURFLOW learns token-wise reference coefficients to adaptively control reference preservation across spatial locations while steering residual learning toward regions requiring edits. These coefficients further guide an edit-aware adversarial objective that concentrates perceptual supervision on edited regions. Extensive experiments across multiple benchmarks and foundation models demonstrate strong few-step editing performance. With only two sampling steps, SPURFLOW closely approaches the full-step FLUX.1-Kontext teacher, surpasses the Qwen-Image-Edit-2511 teacher on both benchmark overall scores, and approaches the performance of the officially distilled 4-step FLUX.2-Klein model.

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.

1 INTRODUCTION

Image editing aims to modify visual content according to user intent while preserving unrelated content, and has long been a fundamental task in computer vision and graphics. With the rapid development of diffusion and flow-based generative models, image editing has made remarkable progress in controllability and visual fidelity Meng et al. (2022); Hertz et al. (2023); Brooks et al. (2023); Zhang et al. (2023). Recent instruction-guided editing models further enable diverse and complex manipulations while maintaining strong consistency with the reference image. However, these models still rely on iterative denoising or flow-based sampling during inference, requiring multiple sequential model evaluations and resulting in substantial inference cost.

Diffusion distillation has emerged as a promising direction for accelerating diffusion and flow-based generative models. DMD Yin et al. (2024b) and DMD2 Yin et al. (2024a) show that high-quality image generation can be achieved with only a few sampling steps by transferring knowledge from pretrained teachers. More recently, trajectory, distribution, and imitation based distillation methods have extended this paradigm to large-scale flow-based text-to-image models, achieving competitive few-step generation quality Luo et al. (2025); Ge et al. (2026); Chen et al. (2026b). Despite different formulations, they share a common principle: training a few-step student to reproduce a multi-step teacher by matching its scores, velocity fields, distributions, or trajectories.

Although few-step distillation has achieved remarkable success in text-to-image generation, extending it to image editing remains challenging. Recent works have begun to explore few-step image editing through diffusion and flow distillation Zhao et al. (2026); Cheng et al. (2026); Wei et al. (2026). However, reducing instruction-based image editing to only a few sampling steps while retaining the quality of the original multi-step model remains difficult. This raises a natural question: why is editing so difficult to distill, even though it typically modifies only part of an existing image rather than generating the entire image from noise?

The answer emerges when we revisit the essence of image editing. Unlike image generation, editing is inherently asymmetric: for most editing tasks, the majority of the reference content should remain unchanged, while only a small subset requires modification. Existing distillation objectives largely ignore this asymmetry and train the student to reproduce the teacher's full score or velocity field uniformly. As illustrated in Fig. 2, this imbalance manifests in two aspects. First, preserved tokens substantially outnumber edited tokens and are typically easier to learn. Second, their gradients dominate the overall training update, further diluting the sparse but critical edit signals. Consequently, the student is biased toward preservation while the regions that actually require modification are insufficiently learned. This suggests that the bottleneck of few-step editing lies not only in the sampling budget, but also in a distillation target that fails to focus on what actually needs to change.

Motivated by this insight, we propose SPURFLOW, a few-step distillation framework that performs **S**patially-adaptive **P**reservation through **U**nsupervised localization and **R**esidual steering for flow matching. Instead of reproducing the teacher's full velocity field, SPURFLOW learns token-wise reference coefficients to preserve reference content while steering residual updates toward regions requiring modification. These coefficients emerge without explicit edit-mask supervision and further guide an edit-aware adversarial objective that concentrates perceptual supervision on edited regions. Together, these designs enable high-quality two-step editing with performance comparable to the original multi-step model. Our contributions are summarized as follows:

- **Conceptually**, we identify a fundamental mismatch between conventional distillation objectives and the asymmetric nature of image editing. Rather than uniformly reproducing the teacher's full generation field, we argue that few-step editing should focus on the sparse but critical editing residual while preserving unchanged content.
- **Technically**, we propose SPURFLOW, an editing-oriented distillation framework that learns spatially varying reference coefficients without explicit supervision to distinguish edited and preserved regions. Building on this decomposition, we further introduce an edit-aware adversarial objective that concentrates learning on regions requiring modification.
- **Empirically**, extensive experiments across multiple benchmarks and foundation editing models demonstrate that SPURFLOW consistently narrows the gap between two-step editing and the original multi-step models, achieving comparable performance and even surpassing them in several editing scenarios.

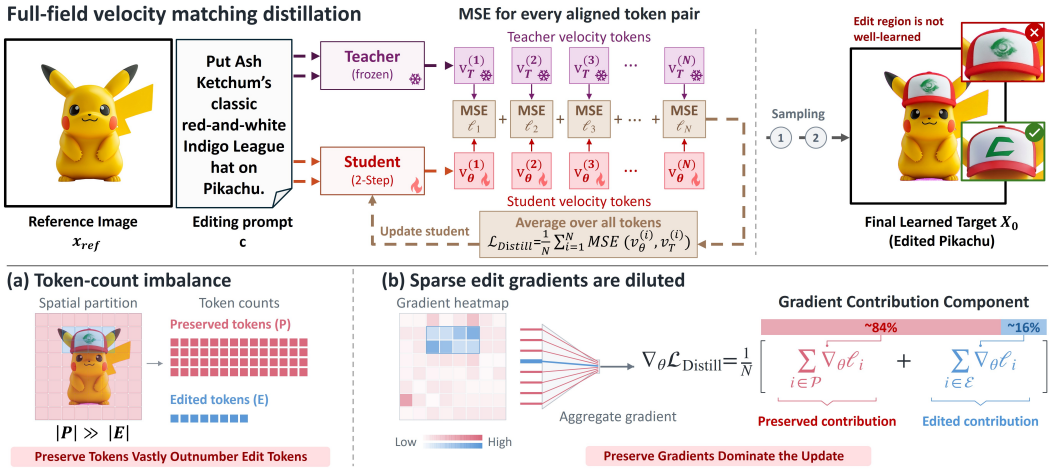

Figure 2: Full-field velocity matching uniformly supervises all image tokens and can produce inaccurate or blurry edits. (a) Preserved tokens vastly outnumber edited tokens, and (b) preserved regions dominate the gradient contribution. The approximately 84%/16% split reports the gradient contributions of the marked preserved/edited regions, measured over all trainable LoRA parameters during FLUX.1-Kontext training and averaged over iterations 1k–20k.

2 Related Work

2.1 Diffusion and Flow-based Image Editing

Diffusion models have enabled substantial advances in image editing through strong generative priors and flexible conditioning. Early approaches primarily manipulate the sampling process of pretrained diffusion models to preserve source content while achieving desired edits. SDEdit Meng et al. (2022) perturbs the input image with noise and performs conditional denoising, while Prompt-to-Prompt Hertz et al. (2023) controls semantic changes through cross-attention manipulation. Null-text Inversion Mokady et al. (2023) further improves real-image editing by optimizing null-text embeddings for better source reconstruction.

Recent works increasingly adopt instruction-guided editing models that perform edits directly from natural-language instructions. InstructPix2Pix Brooks et al. (2023) trains a conditional diffusion model for instruction-based editing without test-time optimization. MagicBrush Zhang et al. (2023), UltraEdit Zhao et al. (2024), and Pico-Banana-400K Qian et al. (2026) further improve instruction-based editing through larger and more diverse training data and supervision.

More recent foundation editing models, such as FLUX.1-Kontext Black Forest Labs et al. (2025) and Qwen-Image-Edit Wu et al. (2025), achieve strong editing quality and instruction following using large pretrained generative models and multimodal understanding. RegionE Chen et al. (2026c) exploits spatial edit sparsity for region-aware acceleration, while Rethinking Where to Edit (RWTE) He et al. (2026) introduces task-aware localization to preserve irrelevant regions. However, these models still rely on iterative denoising or flow sampling, resulting in substantial inference cost. While prior works primarily focus on improving editing quality and controllability, we study efficient image editing through few-step model distillation, raising the question of whether distillation objectives developed for image generation are equally suitable for image editing.

2.2 Diffusion and Flow Matching Distillation

Few-step distillation has been widely explored to accelerate diffusion and flow-based generative models by transferring knowledge from pretrained models to efficient student models. Existing approaches can be broadly categorized into trajectory-based and distribution-based methods.

Trajectory-based methods aim to preserve the sampling dynamics of the teacher model. Direct Distillation Luhman & Luhman (2021) and Progressive Distillation Salimans & Ho (2022) progressively reduce sampling steps by matching intermediate denoising trajectories. Consistency Mod-

els Song et al. (2023) and Latent Consistency Models Luo et al. (2023) enable few-step generation by enforcing consistency across different points along the diffusion trajectory. Recent methods extend this idea to flow-based models. ArcFlow Yang et al. (2026) and Teacher-Aligned End-to-End Distillation Liu et al. (2026) achieve high-quality few-step generation by aligning student and teacher flow dynamics, while Trajectory Distribution Matching Luo et al. (2025) further combines trajectory supervision with distribution alignment.

Distribution-based methods align the student and teacher output distributions. Diffusion Distribution Matching (DMD) Yin et al. (2024b) introduces a score-based distribution matching objective for one-step generation, while DMD2 Yin et al. (2024a) improves stability and generation quality. Adversarial methods, including ADD Sauer et al. (2024b) and LADD Sauer et al. (2024a), enhance perceptual quality through adversarial supervision. More recently, SenseFlow Ge et al. (2026) scales distribution matching distillation to large flow-based text-to-image models.

Despite strong progress in few-step generation, most existing distillation objectives are designed for settings where the entire image is synthesized from noise. Image editing is inherently asymmetric: most reference content should be preserved, while only sparse instruction-specified regions require modification. Under conventional full-field matching, preservation tokens dominate spatially and can consequently dominate the training gradients, diluting the sparse yet critical edit signals. In this work, we revisit diffusion distillation from the perspective of image editing and develop an editing-oriented framework that separates reference-preserving content from edit-specific updates.

3 PRELIMINARIES

We next introduce the flow matching formulation underlying our method and then revisit the distillation methods used for accelerating generative models.

Flow Matching. Given an image sample $I_0 \sim p_{\text{data}}$, we first encode it into the latent space using a pretrained image encoder $\mathcal{E}$, yielding $x_0 = \mathcal{E}(I_0) \in \mathbb{R}^D$. With Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, flow matching defines a continuous interpolation between the data latent and noise as $x_t = \alpha_t x_0 + \sigma_t \epsilon$, where $t \in [0, 1]$ denotes the flow time. For the commonly used linear flow schedule, $\alpha_t = 1 - t$ and $\sigma_t = t$, the interpolation becomes $x_t = (1-t)x_0 + t\epsilon$, and the corresponding sample velocity along this interpolation path is given by

$$u := \frac{dx_t}{dt} = \frac{x_t - x_0}{\sigma_t} = \epsilon - x_0. \tag{1}$$

The corresponding ODE velocity field is given by the conditional expectation of the sample velocity given x_t, i.e., $\frac{dx_t}{dt} = \mathbb{E}_{x_0 \sim p(x_0|x_t)}[u]$. A flow model $v_\theta(x_t, t)$ is trained to approximate this velocity field with the standard flow matching objective $\mathcal{L}_\theta = \mathbb{E}_{t \sim p(t),\, x_0,\, \epsilon}\left[\|v_\theta(x_t, t) - (\epsilon - x_0)\|_2^2\right]$. The learned velocity field is then integrated through an ODE to transport Gaussian noise to the data latent, which is decoded back to image space by the pretrained decoder $\mathcal{D}$.

Few-Step Distillation. Few-step distillation accelerates a pretrained multi-step generative model by training a student with a shorter sampling trajectory. Existing approaches broadly include trajectory-based methods, which approximate teacher sampling dynamics, and distribution-matching methods, which align generated distributions (Luo et al., 2025). Within the flow matching framework, we consider a velocity-matching formulation for trajectory-based distillation (Yang et al., 2026), where the student velocity parameterization enables analytical integration over sampling intervals. The resulting objective aligns student and teacher velocities at sampled intermediate states along the denoising trajectory:

$$\mathcal{L}_{\text{Distill}} = \mathbb{E}_{t_i, x_{t_i}}\left[\|v_\theta(x_{t_i}, t_i) - v_{\text{T}}(x_{t_i}, t_i)\|_2^2\right], \tag{2}$$

where v_θ and v_{T} denote the student and teacher instantaneous velocities, respectively. While such full-field velocity matching is effective for few-step generation, it is not naturally aligned with image editing. Editing is inherently asymmetric: most reference content should remain unchanged, while only sparse instruction-relevant content requires modification. As a result, treating all image tokens uniformly can cause the distillation objective to be dominated by preservation, weakening the learning signal associated with the desired edits. We revisit this mismatch in the next section and develop an editing-specific velocity parameterization.

4 SPURFLOW DISTILLATION

To address the mismatch between conventional full-field distillation and the asymmetric nature of image editing, we propose SPURFLOW. Our key observation is that, unlike image generation, an editing target does not need to be synthesized entirely from scratch: most of its content is already provided by the reference image. This motivates us to explicitly separate reference-preserving content from edit-specific updates in the student velocity. We first introduce reference-aware velocity distillation in Sec. 4, and then extend it with adaptive token-wise reference injection and edit-aware adversarial learning in Sec. 4.

4.1 EDIT-AWARE VELOCITY DISTILLATION

Starting from conventional velocity matching, we reconsider the student parameterization for image editing. Given a reference image I_{ref} with latent representation $x_{\text{ref}} = \mathcal{E}(I_{\text{ref}})$, an editing instruction c, and the corresponding target latent x_0, we write $x_0 = x_{\text{ref}} + \Delta x$, where Δx denotes the instruction-specific modification. Under the linear flow parameterization, the corresponding sample-path velocity is $\epsilon - (x_{\text{ref}} + \Delta x)$. Since x_{ref} is already available to the student, we explicitly incorporate it into the student prediction and parameterize the remaining component with a reference-conditioned residual predictor r_θ. Accordingly, the teacher–student velocity matching objective in Eq. 2 is written as

$$\mathcal{L}_{\text{SpurFlow}} = \mathbb{E}_{t_i, x_{t_i}, \epsilon}\left[\left\|(\epsilon - (x_{\text{ref}} + r_\theta(x_{t_i}, t_i, x_{\text{ref}}, c))) - v_{\text{T}}(x_{t_i}, t_i, x_{\text{ref}}, c)\right\|_2^2\right]. \tag{3}$$

The residual predictor r_θ is supervised through the teacher velocity and need not equal the paired image difference Δx at intermediate states. The reference branch provides an inductive bias for reusing source content. By reducing the need to relearn the reference-preserving component, the student can devote more of its limited few-step capacity to sparse instruction-dependent modifications.

Unsupervised Edit Localization. The reference-aware parameterization in Sec. 4.1 explicitly incorporates the reference image into the student prediction. However, assigning the same reference contribution to all image tokens does not reflect the spatially selective nature of editing: most regions should closely preserve the reference, while only instruction-relevant regions need to deviate from it. We therefore replace the fixed reference contribution with a learnable token-wise coefficient $\alpha_\theta^{(j)}$, allowing the student to adaptively control reference reuse at each spatial location based on the required degree of modification. Accordingly, Eq. 3 becomes

$$\mathcal{L}_{\text{SpurFlow}} = \mathbb{E}_{t_i, x_{t_i}, \epsilon}\left[\sum_{j=1}^{N}\left\|\left(\epsilon^{(j)} - \alpha_\theta^{(j)} x_{\text{ref}}^{(j)} - r_\theta^{(j)}(x_{t_i}, t_i, x_{\text{ref}}, c)\right) - v_{\text{T}}^{(j)}(x_{t_i}, t_i, x_{\text{ref}}, c)\right\|_2^2\right]. \tag{4}$$

Here, $\alpha_\theta^{(j)}$ controls the reference contribution at token j, while $r_\theta^{(j)}$ predicts the remaining edit-specific residual. We predict α_θ using an additional head attached to the shared student backbone, enabling the amount of reference reuse to be adapted independently across image tokens. The complete student architecture is illustrated in Fig. 6 in Appendix C.5.

For the residual prediction r_θ, we retain the multi-mode parameterization of the underlying few-step student Yang et al. (2026):

$$r_\theta^{(j)}(x_t, t, x_{\text{ref}}, c) = \sum_{k=1}^{K} \pi_{\theta,k}(x_t, t, x_{\text{ref}}, c)\, r_{\theta,k}^{(j)}(x_t, t, x_{\text{ref}}, c)\, \gamma_k(x_t)^{1-t}, \tag{5}$$

where $\pi_{\theta,k} = p_\theta(z = k \mid x_t, t, x_{\text{ref}}, c)$ denotes the probability of the k-th mode and $r_{\theta,k}^{(j)}$ denotes its mode-specific residual prediction. Thus, our modification leaves the underlying residual parameterization unchanged and introduces the token-wise coefficient $\alpha_\theta^{(j)}$ to adaptively control reference reuse. Appendix B provides a token-wise gradient analysis explaining how this parameterization can alleviate the gradient imbalance illustrated in Fig. 2(b). Although α_θ receives no explicit edit-mask or spatial supervision, its learned coefficients naturally align with regions requiring modification. This emergent spatial structure provides an unsupervised edit-localization signal, which we use next to guide localized adversarial supervision.

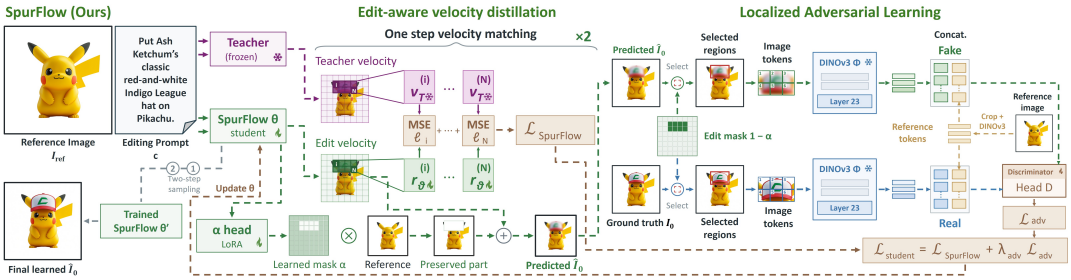

Figure 3: **Overview of SPURFLOW.** SPURFLOW learns token-wise reference coefficients to adaptively control reference reuse while predicting edit-specific residuals during velocity distillation. The learned coefficients further provide an implicit edit mask, which selects edit-relevant regions for localized adversarial supervision. The student is trained with two-step rollout and jointly optimized by the distillation and adversarial objectives.

Localized Adversarial Learning. While velocity distillation constrains the student dynamics, accurately recovering fine-grained modifications remains difficult under an extremely short sampling trajectory. We therefore use the edit localization induced by α_θ to concentrate additional adversarial supervision on regions requiring modification. Specifically, we decode the clean student prediction into an image $\hat{I}_0 = \mathcal{D}(\hat{x}_0)$, where $\mathcal{D}$ denotes the image decoder. From the learned token-wise coefficients, we derive an editing region $\mathcal{R}_\alpha$ in image coordinates and crop the same spatial region from the reference image I_{ref}, the paired edited target I_0, and the decoded student image $\hat{I}_0$.

A frozen DINOv3 encoder Φ extracts features from these image crops. To make the discriminator sensitive to the modification relative to the reference, we concatenate the reference features with either the target or student features, forming real and fake inputs to the discriminator head D:

$$\begin{aligned} f_{\text{real}} &= \left[\Phi(\mathcal{C}(I_{\text{ref}}; \mathcal{R}_\alpha)), \Phi(\mathcal{C}(I_0; \mathcal{R}_\alpha))\right], \\ f_{\text{fake}} &= \left[\Phi(\mathcal{C}(I_{\text{ref}}; \mathcal{R}_\alpha)), \Phi\Big(\mathcal{C}(\hat{I}_0; \mathcal{R}_\alpha)\Big)\right], \end{aligned} \tag{6}$$

where $\mathcal{C}(\cdot; \mathcal{R}_\alpha)$ denotes cropping at the localized editing region, and $[\cdot, \cdot]$ denotes channel-wise feature concatenation. Conditioning on the same reference crop encourages D to assess whether the predicted modification is plausible relative to the source content. With D outputting logits, we optimize the discriminator and student using

$$\begin{aligned} \mathcal{L}_D &= \mathbb{E}\left[\text{softplus}(D(f_{\text{fake}})) + \text{softplus}(-D(f_{\text{real}}))\right], \\ \mathcal{L}_{\text{adv}} &= \mathbb{E}\left[\text{softplus}(-D(f_{\text{fake}}))\right]. \end{aligned} \tag{7}$$

We average these losses over views and minimize $\mathcal{L}_{\text{student}} = \mathcal{L}_{\text{SpurFlow}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}}$.

4.2 TRAINING AND INFERENCE

Training. We first warm up the student by optimizing $\mathcal{L}_{\text{SpurFlow}}$ over one randomly sampled interval, stabilizing the residual predictor r_θ and token-wise reference coefficients α_θ. We then perform a complete student rollout through both intervals, retaining the same distillation objective within each interval and applying the localized adversarial loss $\mathcal{L}_{\text{adv}}$ to the decoded student image $\hat{I}_0 = \mathcal{D}(\hat{x}_0)$.

Inference. Starting from Gaussian noise, the student follows the same 2-NFE rollout used during training, and the final latent is decoded into the edited image. Detailed training and inference procedures are provided in Appendix A.

5 EMPIRICAL ANALYSIS

We conduct extensive experiments to evaluate the effectiveness of SPURFLOW through the following three guiding questions. **Q1: How does SPURFLOW compare with existing few-step editing methods?** We compare SPURFLOW with existing few-step editing methods and original multi-step teacher models on standard image editing benchmarks, assessing both editing quality and efficiency

Table 1: Quantitative comparisons on ImgEdit-Bench and GEdit-Bench v2. All models are evaluated using the same evaluation protocol. Best and second-best overall results among distilled models within each backbone group are highlighted in bold and underlined, respectively.

Idx	Method	NFE	ImgEdit-Bench↑										GEdit-Bench v2↑		
			Action	Add	Adjust	Background	Compose	Extract	Remove	Replace	Style	Average	Q-SC	Q-PQ	Q-O
Base: FLUX.1-Kontext (2025)		28	4.51	4.15	3.97	4.06	3.09	2.33	3.26	4.50	4.58	3.83	6.36	7.98	6.12
①	ArcFlow (2026)	2	3.80	3.73	3.91	3.60	3.03	2.14	3.07	4.01	3.97	3.45	5.87	7.66	<u>5.59</u>
②	SenseFlow (2026)	2	3.66	3.77	4.27	3.87	3.49	1.73	2.89	3.93	4.62	<u>3.57</u>	5.60	7.33	5.27
③	UniPic 3.0 (2026)	4	3.12	3.44	3.28	3.02	2.58	1.95	2.57	3.02	4.01	3.01	5.81	6.44	5.46
④	DMD2 (2024a)	4	3.92	3.95	3.69	3.05	3.09	1.81	2.31	3.26	3.96	3.17	5.74	7.27	5.39
⑤	**SpurFlow (Ours)**	2	4.27	4.08	4.19	4.03	3.49	2.52	3.55	4.17	4.02	**3.81**	6.10	8.03	**5.93**
Base: Qwen-Image-Edit-2511 (2025)		40×2	4.66	4.55	4.45	4.35	3.59	4.13	4.41	4.63	4.87	4.46	8.07	8.17	7.74
①	ArcFlow (2026)	2	4.75	4.33	4.44	4.19	3.36	3.60	4.49	4.59	4.90	4.33	8.20	7.89	<u>7.76</u>
②	SenseFlow (2026)	2	3.53	3.15	4.56	3.85	3.49	2.73	4.14	3.63	4.61	3.76	7.03	6.82	6.48
③	TwinFlow (2026)	4	–	–	–	–	–	–	–	–	–	3.55	5.95	6.97	5.91
④	Qwen-Image-Edit-2511-Lightning (2025)	4	4.75	4.41	4.37	4.17	3.71	4.07	4.35	4.63	4.88	4.40	8.09	8.10	7.74
⑤	TeleStyleV2-QIE-2511-DMD (2026)	4	4.65	4.45	4.59	4.25	3.75	3.94	4.37	4.65	4.96	<u>4.44</u>	8.01	8.35	<u>7.76</u>
⑥	**SpurFlow (Ours)**	2	4.66	4.71	4.60	4.52	4.01	3.40	4.76	4.83	4.77	**4.48**	8.33	8.10	**7.94**
Base: FLUX.2-Klein-9B-Base (2026)		50×2	4.98	4.58	4.34	4.43	3.85	2.51	3.95	4.67	4.89	4.21	7.81	8.19	7.49
①	ArcFlow (2026)	2	4.71	4.33	4.36	4.36	3.68	2.18	3.44	4.23	4.82	3.97	7.36	8.05	7.09
②	SenseFlow (2026)	2	3.47	3.44	4.34	3.95	3.88	2.11	3.53	3.68	4.86	3.67	6.82	7.26	6.25
③	DMD2 (2024a)	2	3.82	4.12	3.70	3.56	3.01	2.19	2.40	3.65	4.56	3.46	7.17	7.40	6.73
④	FLUX.2-Klein-9B (2026)	4	4.85	4.64	4.55	4.44	3.48	2.42	4.63	4.52	4.79	**4.24**	7.84	8.32	**7.58**
⑤	**SpurFlow (Ours)**	2	4.57	4.37	4.64	4.52	3.65	2.23	4.52	4.56	4.85	<u>4.21</u>	7.60	8.08	<u>7.31</u>

across different foundation models. **Q2: Are the proposed designs in SPURFLOW effective?** We perform comprehensive ablation studies to analyze the contribution of each component, including the learnable token-wise reference coefficients and the edit-aware adversarial objective. **Q3: Can SPURFLOW accurately localize edited regions without explicit spatial supervision?** We examine whether the learned token-wise reference coefficients can accurately identify instruction-relevant editing regions without explicit edit-mask supervision. Due to space limitations, details of the datasets, evaluation metrics, and implementation settings are provided in Appendix C.

5.1 COMPARISON WITH EXISTING METHODS

To answer Q1, we compare SPURFLOW with eight few-step baselines and the original multi-step teachers on ImgEdit-Bench and GEdit-Bench v2, as summarized in Table 1. The results demonstrate two main advantages. Across all three foundation models, SPURFLOW outperforms all other evaluated 2-step baselines in both ImgEdit-Bench Average and GEdit-Bench v2 Q-O. With FLUX.1-Kontext, our 2-step student achieves 3.81 and 5.93, respectively, compared with 3.45 and 5.59 for 2-step ArcFlow, and 3.57 and 5.27 for 2-step SenseFlow. It also substantially surpasses 4-step DMD2 and UniPic 3.0 with fewer sampling steps. On Qwen-Image-Edit-2511, SPURFLOW improves over ArcFlow from 4.33 to 4.48 on ImgEdit-Bench and from 7.76 to 7.94 on GEdit-Bench v2. On FLUX.2-Klein-9B-Base, it exceeds all evaluated 2-step baselines on both overall metrics. These gains support editing-oriented distillation that separates reference preservation from edit-specific learning and concentrates perceptual supervision on edited regions.

SPURFLOW also achieves competitive performance against publicly released 4-step models using only two function evaluations. Our Qwen-based student obtains the highest overall scores among the compared distilled models in this group, surpassing Qwen-Image-Edit-2511-Lightning and TeleStyleV2-QIE-2511-DMD on both benchmarks. It also slightly exceeds its multi-step teacher, which scores 4.46 on ImgEdit-Bench and 7.74 on GEdit-Bench v2. For FLUX.2-Klein, our 2-step student closely approaches the official 4-step model on ImgEdit-Bench (4.21 vs. 4.24) and remains competitive on GEdit-Bench v2 (7.31 vs. 7.58). These results show that editing-oriented distillation can achieve strong editing quality with half the sampling budget of existing 4-step releases.

5.2 EFFECTIVENESS OF PROPOSED DESIGNS

To answer **Q2**, we first investigate whether adding the mixed training corpus alone produces a substantial improvement, thereby separating gains due to data composition from those due to the proposed model and training objectives. To this end, we evaluate an SFT baseline trained on the same mixed training corpus described in Appendix C.1. As shown in Table 2, this configuration improves the ImgEdit-Bench average from 3.83 to 3.89, while reducing the overall VIEScore on GEditBench v2 from 6.12 to 5.66. This indicates that additional editing data alone cannot explain the gains achieved by SPURFLOW and may even reduce generalization.

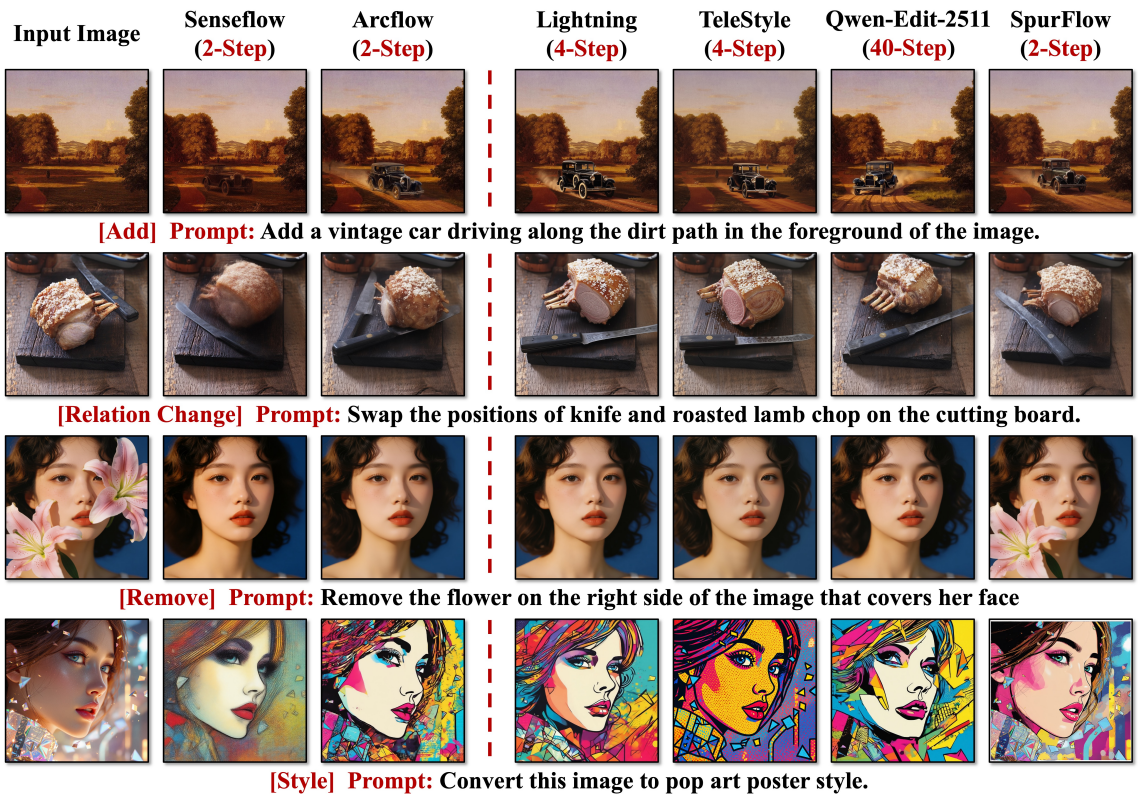

Figure 4: **Qualitative comparison.** SPURFLOW clearly outperforms 2-step baselines, with particularly noticeable improvements in object removal, and remains competitive with strong 4-step methods using only 2 steps. More qualitative examples are provided in Appendix F.

Table 2: Data and component ablations of SPURFLOW on ImgEdit-Bench and GEdit-Bench v2. Rows ③ and ④ both retain the reference term: the former uses a fixed coefficient, while the latter learns token-wise coefficients α. Global GAN uses only full-image supervision. Random-Crop GAN adds randomly sampled local crops, while α-Guided GAN uses the learned α to focus local supervision on predicted edit regions. Both local variants retain global supervision.

Idx	Method	NFE↓	ImgEdit-Bench↑										GEdit-Bench v2↑		
			Action	Add	Adjust	Background	Compose	Extract	Remove	Replace	Style	Average	Q-SC	Q-PQ	Q-O
Data Ablation															
①	FLUX.1-Kontext	28	4.51	4.15	3.97	4.06	3.09	2.33	3.26	4.50	4.58	3.83	6.36	7.98	6.12
②	① + SFT (Same Data as SPURFLOW)	28	4.74	4.22	4.10	4.13	3.51	2.28	3.35	4.48	4.59	3.89	5.81	7.30	5.66
Component Ablation															
③	w/o Learnable Reference Coefficients	2	3.95	3.72	3.85	3.55	2.83	2.30	3.25	4.02	4.01	3.49	6.01	7.65	5.72
④	w/ Learnable Reference Coefficients	2	4.13	3.87	3.91	3.69	2.90	2.26	3.58	4.10	3.97	3.59	6.03	7.70	5.76
⑤	④ + Global GAN	2	4.04	3.96	3.87	3.66	3.38	2.41	3.44	4.18	3.98	3.63	6.01	7.80	5.78
⑥	④ + Random-Crop GAN	2	3.71	3.81	4.12	3.76	3.03	2.05	3.84	4.19	3.94	3.62	5.99	7.79	5.77
⑦	④ + α-Guided GAN (Ours)	2	4.27	4.08	4.19	4.03	3.49	2.52	3.55	4.17	4.02	**3.81**	6.10	8.03	**5.93**

We evaluate the components under the same 2-step setting. Learning token-wise reference coefficients within our edit-aware velocity formulation improves the ImgEdit-Bench average from 3.49 to 3.59 and GEdit-Bench v2 Q-O from 5.72 to 5.76, demonstrating consistent gains from explicitly modeling spatially varying reference preservation. Using the learned edit regions to guide adversarial supervision further improves performance on both editing benchmarks. Global GAN yields an ImgEdit-Bench average of 3.63 and GEdit-Bench v2 Q-O of 5.78, while Random-Crop GAN achieves scores of 3.62 and 5.77. Our localized adversarial objective improves these scores to 3.81 and 5.93, respectively, with Q-SC reaching 6.10 and Q-PQ reaching 8.03. Relative to the variant with learned reference coefficients but no adversarial supervision, this corresponds to additional gains of 0.22 on ImgEdit-Bench and 0.17 on GEdit-Bench v2 Q-O. Its improvements over both global and random-crop supervision support the benefit of using learned edit localization to concentrate perceptual supervision on instruction-relevant regions.

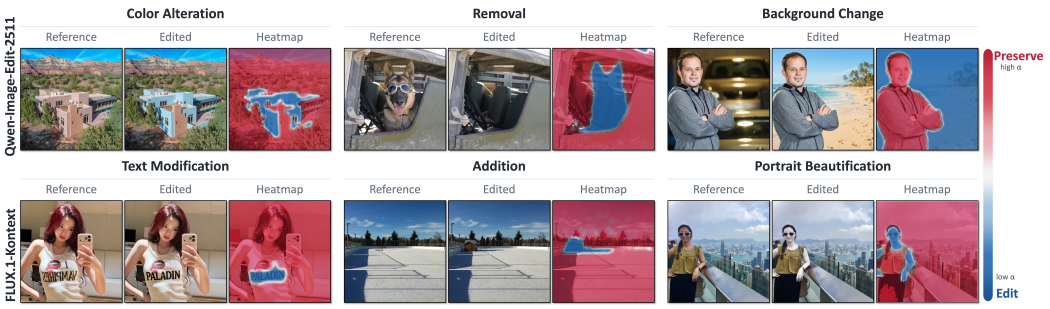

Figure 5: **Visualization of unsupervised edit localization.** Heatmaps derived from the learned token-wise reference coefficients of SPURFLOW on Qwen-Image-Edit-2511 and FLUX.1-Kontext. Without explicit edit-mask supervision, the learned localization aligns well with instruction-relevant regions across diverse editing categories.

Table 3: Edit-region localization results on 507 PIE-Bench samples. We report category-wise and overall mask IoU against human annotations. The GPT-5.5 score (0–10) is the unweighted mean of semantic alignment, coverage, precision, and compactness. All methods are based on FLUX.1-Kontext-dev.

Method	Localization IoU↑									GPT-5.5↑
	Random	Replace Obj.	Delete Obj.	Attr. Content	Attr. Pose	Attr. Color	Attr. Material	Background	Overall	Average
RegionE(2026c)	0.470	0.383	0.373	0.327	**0.368**	0.569	0.493	0.631	0.455	5.191
RWTE(2026)	0.359	0.366	0.276	0.321	0.318	0.383	0.458	0.587	0.384	4.721
SPURFLOW	**0.535**	**0.455**	**0.394**	**0.377**	0.318	**0.631**	**0.499**	**0.670**	**0.496**	**5.454**

5.3 UNSUPERVISED EDIT LOCALIZATION

To answer **Q3**, we examine whether the token-wise reference coefficients learned by SPURFLOW identify instruction-relevant edit regions without explicit mask supervision. As shown in Fig. 5, the learned localization maps highlight regions requiring modification across different foundation models and editing tasks. They capture localized edits, including object addition and removal, attribute changes, and text modification, as well as larger-scale operations such as background replacement and portrait retouching.

We quantitatively evaluate edit-region localization on a 507-sample subset of PIE-Bench Ju et al. (2024). As shown in Table 3, SPURFLOW achieves the highest overall IoU (0.496) and leads in seven of the eight reported categories compared with RegionE Chen et al. (2026c) and our local reproduction of RWTE He et al. (2026). We further use GPT-5.5 to evaluate the predicted masks in terms of semantic alignment, coverage, precision, and compactness, given the source image and editing instruction. The final GPT-based score is the unweighted mean of these four metrics. SPURFLOW achieves the highest average score of 5.454, compared with 5.191 for RegionE and 4.721 for RWTE. Together, these results indicate that the learned reference coefficients provide meaningful spatial cues for distinguishing regions requiring edits from those to be preserved. Details of the evaluation subset and scoring protocol are provided in Appendix E.

6 CONCLUSION

In this work, we presented SPURFLOW, a few-step distillationframework designed for the spatially asymmetric nature ofinstruction-based image editing. SPURFLOW introduces edit-aware velocity distillation with token-wise reference coefficients, enabling adaptive reference preservation and edit-specific residual prediction. The learned coefficients further provide unsupervised localization of instruction-relevant edit regions, which enables localized adversarial supervision without explicit mask annotations. Across three foundation models and multiple benchmarks, SPURFLOW consistently improves over the evaluated general-purpose distillation baselines using only two function evaluations, while remaining competitive with strong four-step editing models and multi-step teachers. Overall, our results suggest that explicitly separating reference preservation from edit-specific changes is an effective inductive bias for few-step image editing distillation.

AI USE STATEMENT

We used ChatGPT to assist with language polishing, manuscript organization, LaTeX formatting, and the preparation and refinement of scientific illustrations. We also used it to support literature search and the identification of relevant prior work. GPT-based models were used for automated evaluation of image editing results and predicted edit masks, as detailed in the evaluation protocols. The authors reviewed and revised the AI-assisted text and illustrations to ensure accuracy and consistency with the method and reported experimental results. The authors take full responsibility for the final content of this paper.

ETHICS STATEMENT

SPURFLOW enables efficient instruction-based image editing, which may be misused to create misleading imagery or modify personal images without consent. The distilled models may also inherit biases, failure modes, and safety limitations from their base models and training data. We encourage responsible use that respects privacy, consent, intellectual property rights, and applicable policies and regulations.

REPRODUCIBILITY STATEMENT

We provide the implementation, training and inference configurations, evaluation scripts, and instructions necessary to reproduce the main experiments of this work in our anonymous code repository: `https://anonymous.4open.science/r/SpurFlow-154F/`. Additional implementation details, including model architecture, training settings, and evaluation protocols, are documented in the appendix and the accompanying repository.

REFERENCES

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. *arXiv preprint arXiv:2506.15742*, 2025.

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*, 2023.

Guanzhou Chen, Erfei Cui, Changyao Tian, Danni Yang, Ganlin Yang, Yu Qiao, Hongsheng Li, Gen Luo, and Hongjie Zhang. Scaleedit-12m: Scaling open-source image editing data generation via multi-agent framework. *arXiv preprint arXiv:2603.20644*, 2026a.

Hansheng Chen, Kai Zhang, Hao Tan, Leonidas Guibas, Gordon Wetzstein, and Sai Bi. π-flow: Policy-based few-step generation via imitation distillation. In *International Conference on Learning Representations*, 2026b.

Pengtao Chen, Xianfang Zeng, Maosen Zhao, Mingzhu Shen, Peng Ye, Bangyin Xiang, Zhibo Wang, Wei Cheng, Gang Yu, and Tao Chen. Regione: Adaptive region-aware generation for efficient image editing. In *International Conference on Learning Representations*, 2026c.

Zhenglin Cheng, Peng Sun, Jianguo Li, and Tao Lin. Twinflow: Realizing one-step generation on large models with self-adversarial flows. *International Conference on Learning Representations*, 2026.

Xingtong Ge, Xin Zhang, Tongda Xu, Yi Zhang, Xinjie Zhang, Yan Wang, and Jun Zhang. Seneflow: Scaling distribution matching for flow-based text-to-image distillation. In *International Conference on Learning Representations*, 2026.

Jingxuan He, Xiyu Wang, Mengyu Zheng, Xiangyu Zeng, Yunke Wang, and Chang Xu. Rethinking where to edit: Task-aware localization for instruction-based image editing. *arXiv preprint arXiv:2604.20258*, 2026.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *IEEE/CVF International Conference on Computer Vision*, 2015.

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

Zhangqi Jiang, Zheng Sun, Xianfang Zeng, Yufeng Yang, Xuanyang Zhang, Yongliang Wu, Wei Cheng, Gang Yu, Xu Yang, and Bihan Wen. Geditbench v2: A human-aligned benchmark for general image editing. *arXiv preprint arXiv:2603.28547*, 2026.

Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. PnP inversion: Boosting diffusion-based editing with 3 lines of code. In *International Conference on Learning Representations*, 2024.

Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. Viescore: Towards explainable metrics for conditional image synthesis evaluation. In *Annual Meeting of the Association for Computational Linguistics*, 2024.

Black Forest Labs. Flux.2 [klein] 9b base. `https://huggingface.co/black-forest-labs/FLUX.2-klein-base-9B`, 2026.

Jie Liang, Hui Zeng, Miaomiao Cui, Xuansong Xie, and Lei Zhang. Ppr10k: A large-scale portrait photo retouching dataset with human-region mask and group-level consistency. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021.

LightX2V. Qwen-image-edit-2511-lightning. `https://huggingface.co/lightx2v/Qwen-Image-Edit-2511-Lightning`, 2025. 4-step distilled LoRA for Qwen-Image-Edit-2511.

Dongyang Liu, Ruoyi Du, David Liu, Dengyang Jiang, Liangchen Li, Qilong Wu, Zhen Li, Steven C. H. Hoi, Hongsheng Li, and Peng Gao. High-fidelity two-step image generation via teacher-aligned end-to-end distillation. *arXiv preprint arXiv:2606.12575*, 2026.

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. *arXiv preprint arXiv:2504.17761*, 2025.

Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Yihong Luo, Tianyang Hu, Jiacheng Sun, Yujun Cai, and Jing Tang. Learning few-step diffusion models by trajectory distribution matching. In *IEEE/CVF International Conference on Computer Vision*, 2025.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

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

Yusu Qian, Eli Bocek-Rivele, Liangchen Song, Jialing Tong, Yinfei Yang, Jiasen Lu, Wenze Hu, and Zhe Gan. Pico-banana-400k: A large-scale dataset for text-guided image editing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Qwen Team. Qwen-image-edit-2511. https://huggingface.co/Qwen/Qwen-Image-Edit-2511, 2025.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *SIGGRAPH Asia*, 2024a.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, 2024b.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, 2023.

Hongyang Wei, Hongbo Liu, Zidong Wang, Yi Peng, Baixin Xu, Size Wu, Xuying Zhang, Xianglong He, Zexiang Liu, Peiyu Wang, Xuchen Song, Yangguang Li, Yang Liu, and Yahui Zhou. Skywork unipic 3.0: Unified multi-image composition via sequence modeling, 2026. URL https://arxiv.org/abs/2601.15664.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Donglu Yang, Liang Zhang, Zihao Yue, Liangyu Chen, Yichen Xu, Wenxuan Wang, and Qin Jin. Chartm3: Benchmarking chart editing with multimodal instructions. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 2025.

Zihan Yang, Shuyuan Tu, Licheng Zhang, Qi Dai, Yu-Gang Jiang, and Zuxuan Wu. Arcflow: Unleashing 2-step text-to-image generation via high-precision non-linear flow distillation. *arXiv preprint arXiv:2602.09014*, 2026.

Keming Ye, Zhipeng Huang, Canmiao Fu, Qingyang Liu, Jiani Cai, Zheqi Lv, Chen Li, Jing Lyu, Zhou Zhao, and Shengyu Zhang. Unicedit-10m: A dataset and benchmark breaking the scale-quality barrier via unified verification for reasoning-enriched edits. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

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 *Advances in Neural Information Processing Systems*, 2025. Datasets and Benchmarks Track.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems*, 2024a.

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*, 2024b.

Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. In *Advances in Neural Information Processing Systems*, 2023.

Shiwen Zhang, Yifan Xu, Haibin Huang, Chi Zhang, and Xuelong Li. Telestyle v2: Beyond content-preserving style transfer with self-distillation and distribution-matching-distillation. *arXiv preprint arXiv:2606.20709*, 2026.

Chuyang Zhao, Yifei Song, Hongfa Wang, Jianlong Yuan, Yuan Zhang, Siming Fu, Zhineng Chen, Huilin Deng, Haoyang Huang, and Nan Duan. Perceptual flow matching for few-step generative modeling. *arXiv preprint arXiv:2607.03524*, 2026.

Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale. In *Advances in Neural Information Processing Systems*, 2024.

A TRAINING AND INFERENCE ALGORITHMS

We provide the training and inference procedures of SPURFLOW. Training begins with interval-based warm-up using only $\mathcal{L}_{\mathrm{SpurFlow}}$. After warm-up, we perform a complete student rollout, using $\mathrm{NFE} = 2$ by default, applying the same objective within each interval and localized adversarial supervision to the decoded final prediction. Inference follows the same student rollout. The algorithms below are written for general NFE.

Warm-up. At each iteration, we sample one interval and construct its starting latent by forward diffusion of the paired target x_0. We obtain intermediate states x_t using mixed latent integration Yang et al. (2026) and stop gradients through them. The residual predictor r_θ and token-wise reference coefficients α_θ are jointly optimized through teacher–student velocity matching.

Algorithm 1: SPURFLOW Warm-up

Input: student residual predictor r_θ, reference coefficient head α_θ, frozen teacher v_{T}, paired training data, sampling steps NFE, total iterations N
Output: optimized student $(r_\theta, \alpha_\theta)$

for $n \leftarrow 1$ **to** N **do**
- Sample a batch $(x_{\mathrm{ref}}, x_0, c)$
- Sample $t_{\mathrm{src}} \in \left\{\frac{1}{\mathrm{NFE}}, \ldots, 1\right\}$
- Sample $\epsilon \sim \mathcal{N}(0, I)$ and set $x_{t_{\mathrm{src}}} \leftarrow (1 - t_{\mathrm{src}})x_0 + t_{\mathrm{src}}\epsilon$
- $x_t \leftarrow \mathrm{MixedIntegration}\left(x_{t_{\mathrm{src}}}, \left[t_{\mathrm{src}} - \frac{1}{\mathrm{NFE}}, t_{\mathrm{src}}\right]; \epsilon - \alpha_\theta \odot x_{\mathrm{ref}} - r_\theta, v_{\mathrm{T}}\right)$
- $\bar{x}_t \leftarrow \mathrm{stopgrad}(x_t)$
- $\alpha_t \leftarrow \alpha_\theta(\bar{x}_t, t, x_{\mathrm{ref}}, c)$
- $\mathcal{L}_{\mathrm{SpurFlow}} \leftarrow \mathbb{E}_t\left[\|\epsilon - \alpha_t x_{\mathrm{ref}} - r_\theta(\bar{x}_t, t, x_{\mathrm{ref}}, c) - v_{\mathrm{T}}(\bar{x}_t, t, x_{\mathrm{ref}}, c)\|_2^2\right]$
- Update θ using $\nabla_\theta \mathcal{L}_{\mathrm{SpurFlow}}$

return $(r_\theta, \alpha_\theta)$

SPURFLOW Training. After warm-up, the student performs a complete NFE-step rollout from Gaussian noise. Within each interval, we optimize $\mathcal{L}_{\mathrm{SpurFlow}}$ at detached intermediate states obtained through mixed latent integration. The student rollout itself remains differentiable for adversarial learning.

The final latent is decoded for adversarial supervision on global and α_θ-guided local views. Within each view, the reference, target, and generated images share the same crop window. A frozen DINOv3 encoder extracts their features, and reference features are concatenated along channels with target or generated features to form real and fake pairs by Eq. 6. We alternate discriminator and student updates using Eq. 7, with losses averaged over views. Fake features are detached during discriminator updates, while discriminator parameters are fixed during student updates. The complete training procedure is summarized in Algorithm 2.

Inference. At inference, SPURFLOW performs an NFE-step student rollout, with $\mathrm{NFE} = 2$ by default. At each step, the student predicts the token-wise reference coefficients and residual parameters, then integrates the resulting edit-aware velocity over the current interval. The term $\alpha_\theta x_{\mathrm{ref}}$ provides token-wise reference weighting, while the initial noise ϵ remains fixed throughout the rollout. The final latent is decoded into the edited image. The complete inference procedure is summarized in Algorithm 3.

Algorithm 2: SPURFLOW Training

Input: warmed-up student $(r_\theta, \alpha_\theta)$, frozen teacher v_{T}, discriminator D_ψ, frozen decoder $\mathcal{D}$ and DINOv3 encoder Φ, paired training data, sampling steps NFE, total iterations N, adversarial weight λ_{adv}
Output: optimized student $(r_\theta, \alpha_\theta)$

for $n \leftarrow 1$ **to** N **do**
  Sample a paired batch $(I_{\mathrm{ref}}, I_0, c)$ with corresponding latents (x_{ref}, x_0)
  Sample $\epsilon \sim \mathcal{N}(0, I)$, $\hat{x}_1 \leftarrow \epsilon$
  $\mathcal{L}_{\mathrm{SpurFlow}} \leftarrow 0$
  for $i \leftarrow 1$ **to** NFE **do**
    $t_{\mathrm{src}} \leftarrow 1 - \frac{i-1}{\mathrm{NFE}}$, $t_{\mathrm{dst}} \leftarrow 1 - \frac{i}{\mathrm{NFE}}$
    $x_t \leftarrow \mathrm{MixedIntegration}\left(\hat{x}_{t_{\mathrm{src}}}, [t_{\mathrm{dst}}, t_{\mathrm{src}}]; (r_\theta, \alpha_\theta), v_{\mathrm{T}}\right)$
    $\bar{x}_t \leftarrow \mathrm{stopgrad}(x_t)$
    Accumulate $\mathcal{L}_{\mathrm{SpurFlow}}$ on $\bar{x}_t$ by Eq. 4
    $(\hat{x}_{t_{\mathrm{dst}}}, \alpha^{(i)}_{\mathrm{roll}}) \leftarrow \mathrm{Integrate}\left(\hat{x}_{t_{\mathrm{src}}}, [t_{\mathrm{dst}}, t_{\mathrm{src}}]; (r_\theta, \alpha_\theta)\right)$, retaining rollout gradients
  $\hat{I}_0 \leftarrow \mathcal{D}(\hat{x}_0)$
  Construct global and local training views $\mathcal{V}$, using low values of $\alpha^{(\mathrm{NFE})}_{\mathrm{roll}}$ to locate local regions
  foreach $v \in \mathcal{V}$ **do**
    Apply the same crop window to I_{ref}, I_0, and $\hat{I}_0$
    Extract features with Φ and concatenate reference features with target or generated features to form $f^{(v)}_{\mathrm{real}}$ and $f^{(v)}_{\mathrm{fake}}$ by Eq. 6
  Update ψ using $\mathcal{L}_D$ in Eq. 7, averaged over views and with fake features detached
  With the updated D_ψ fixed, evaluate $\mathcal{L}_{\mathrm{adv}}$ by Eq. 7, averaged over views
  Update θ using $\nabla_\theta(\mathcal{L}_{\mathrm{SpurFlow}} + \lambda_{\mathrm{adv}}\mathcal{L}_{\mathrm{adv}})$
return $(r_\theta, \alpha_\theta)$

Algorithm 3: SPURFLOW Inference

Input: trained student $(r_\theta, \alpha_\theta)$, decoder $\mathcal{D}$, reference x_{ref}, condition c, sampling steps NFE
Output: edited image $\hat{I}_0$

Sample $\epsilon \sim \mathcal{N}(0, I)$, $\hat{x}_1 \leftarrow \epsilon$
for $i \leftarrow 1$ **to** NFE **do**
  $t_{\mathrm{src}} \leftarrow 1 - \frac{i-1}{\mathrm{NFE}}$, $t_{\mathrm{dst}} \leftarrow 1 - \frac{i}{\mathrm{NFE}}$
  Predict α_θ and residual parameters at $(\hat{x}_{t_{\mathrm{src}}}, t_{\mathrm{src}}, x_{\mathrm{ref}}, c)$, and form $r_\theta(\tau)$ following Eq. 5
  $\hat{x}_{t_{\mathrm{dst}}} \leftarrow \hat{x}_{t_{\mathrm{src}}} - \int_{t_{\mathrm{dst}}}^{t_{\mathrm{src}}} \left[\epsilon - (\alpha_\theta x_{\mathrm{ref}} + r_\theta(\tau))\right] \mathrm{d}\tau$
$\hat{I}_0 \leftarrow \mathcal{D}(\hat{x}_0)$
return $\hat{I}_0$

B WHY EDITED REGIONS HAVE GREATER GRADIENT INFLUENCE IN SPURFLOW

The purpose of this analysis is to explain why the loss on edited regions can have a stronger influence on the parameter update in SPURFLOW. The key mechanism is token-wise routing: the coefficient α_j controls how much of the reference feature is reused, while the residual branch accounts for what the reference cannot explain. For an ideally preserved token, $\alpha_j = 1$ removes the reference component from the residual error; for an ideally edited token, $\alpha_j = 0$ exposes the residual branch to the full target discrepancy. We derive this effect from the exact token-wise gradient of the velocity-distillation loss by decomposing it into a reference-reuse term and a remaining-mismatch term, and then evaluating the same decomposition at these two gate endpoints. The analysis is for one supervised sample with fixed time, noise, reference image, instruction, and detached teacher state. Consequently, the teacher output and the detached state are constants with respect to the trainable parameters θ.

Let $j = 1, \ldots, N$ index spatial tokens, let $\theta \in \mathbb{R}^p$ denote the trainable parameters, and let d be the latent feature dimension at each token. Define

$$x_j = x_{\mathrm{ref},j}, \qquad q_j = \epsilon_j - v_{T,j}, \qquad r_j = r_{\theta,j}(z), \tag{8}$$

where z collects the network inputs, including the detached intermediate state and the editing condition. The reference coefficient is part of the network computation graph:

$$s_j = [h_\theta(z)]_j, \qquad \alpha_j = A(s_j), \tag{9}$$

where h_θ produces token-wise coefficient logits and A is a differentiable bounded activation such as sigmoid or Softsign-01. Thus, α_j is not an independent constant parameter.

For one sample, the token-wise velocity error and the loss are

$$e_j = q_j - \alpha_j x_j - r_j, \qquad \mathcal{L}(\theta) = \sum_{j=1}^{N} \|e_j\|_2^2. \tag{10}$$

If a token-mean loss is used instead, all expressions below are divided by N.

Exact gradient for shared parameters. Define the residual Jacobian and coefficient gradient by

$$J_{r,j} = \frac{\partial r_j}{\partial \theta} \in \mathbb{R}^{d \times p}, \qquad a_j = \nabla_\theta \alpha_j \in \mathbb{R}^p. \tag{11}$$

Applying the chain rule to the actual computation graph gives the exact gradient over all tokens:

$$\boxed{G := \nabla_\theta \mathcal{L} = -2 \sum_{j=1}^{N} \left[J_{r,j}^{\mathsf{T}} e_j + \langle e_j, x_j \rangle a_j \right].} \tag{12}$$

The coefficient path must also be differentiated through its logits. With $b_j = \nabla_\theta s_j = \nabla_\theta [h_\theta(z)]_j$, we have

$$a_j = A'(s_j) b_j, \tag{13}$$

and hence, for a general activation,

$$\boxed{G = -2 \sum_{j=1}^{N} \left[J_{r,j}^{\mathsf{T}} e_j + \langle e_j, x_j \rangle A'(s_j) \nabla_\theta [h_\theta(z)]_j \right].} \tag{14}$$

This expression includes both the activation derivative and the Jacobian of the logit-producing network. It does not assume independent token parameters or independent token computations. Attention interactions, mixture weights, residual components, and time-dependent factors are all included in the corresponding Jacobians.

For completeness, let a shared backbone be $H = F_\phi(z)$, with residual and coefficient heads $r_j = R_{\psi_r,j}(H)$ and $s_j = C_{\psi_\alpha,j}(H)$. The gradient with respect to the shared backbone parameters ϕ can then be written as

$$\nabla_\phi \mathcal{L} = -2 J_{F,\phi}^{\mathsf{T}} \sum_{j=1}^{N} \left[(D_H R_j)^{\mathsf{T}} e_j + \langle e_j, x_j \rangle A'(s_j) \nabla_H C_j \right], \tag{15}$$

where $J_{F,\phi} = \partial H / \partial \phi$. This makes explicit that the residual and coefficient branches merge at the shared representation and jointly contribute to the same backbone parameters.

Exact reference–mismatch decomposition. Let

$$\beta_j = 1 - \alpha_j, \qquad \eta_j = q_j - x_j - r_j. \tag{16}$$

Then the token error admits the exact identity

$$e_j = \beta_j x_j + \eta_j. \tag{17}$$

The first term is the part of the reference content not supplied directly by reference reuse, whereas η_j is the remaining prediction mismatch relative to full reference reuse. Substituting Eq. equation 17 into the exact gradient gives

$$G = G_{\text{ref}} + G_{\text{mis}}, \tag{18}$$

$$G_{\text{ref}} = -2\sum_{j=1}^{N}\left[\beta_j J_{r,j}^{\mathsf{T}} x_j + \beta_j A'(s_j)\|x_j\|_2^2 b_j\right], \tag{19}$$

$$G_{\text{mis}} = -2\sum_{j=1}^{N}\left[J_{r,j}^{\mathsf{T}}\eta_j + A'(s_j)\langle\eta_j, x_j\rangle b_j\right]. \tag{20}$$

For the sigmoid activation, $A'(s_j) = \alpha_j(1-\alpha_j) = \alpha_j\beta_j$. Therefore,

$$G_{\text{ref}} = -2\sum_{j=1}^{N}\left[\underbrace{\beta_j J_{r,j}^{\mathsf{T}} x_j}_{\text{linear attenuation}} + \underbrace{\alpha_j\beta_j^2\|x_j\|_2^2 b_j}_{\text{quadratic attenuation}}\right], \tag{21}$$

$$G_{\text{mis}} = -2\sum_{j=1}^{N}\left[J_{r,j}^{\mathsf{T}}\eta_j + \alpha_j\beta_j\langle\eta_j, x_j\rangle b_j\right]. \tag{22}$$

Here, G_{ref} is the gradient contribution associated with the portion of the reference latent not directly reused, $\beta_j x_j = (1-\alpha_j)x_j$. The term G_{mis} is associated with the remaining velocity mismatch $\eta_j = q_j - x_j - r_j$ after full reference reuse and the current residual prediction have been accounted for. These are two algebraic parts of the same gradient, not gradients of separate objectives; neither is by itself an edit-region gradient.

The two endpoint settings give a direct interpretation of the decomposition. For a sigmoid gate, interpret $\alpha_j = 1$ as the idealized preserve setting (full reference reuse) and $\alpha_j = 0$ as the idealized edit setting (no reference reuse); these are the limiting gate values $\alpha_j \to 1$ and $\alpha_j \to 0$. At these limits the sigmoid coefficient-path derivatives vanish, and the token-wise summands in Eqs. equation 21–equation 22 reduce to

$$\alpha_j = 1 \quad \text{(ideal preserve)}: \qquad g_{\text{ref},j} = 0, \quad g_{\text{mis},j} = -2J_{r,j}^{\mathsf{T}}\eta_j, \tag{23}$$

$$\alpha_j = 0 \quad \text{(ideal edit)}: \qquad g_{\text{ref},j} = -2J_{r,j}^{\mathsf{T}}x_j, \quad g_{\text{mis},j} = -2J_{r,j}^{\mathsf{T}}\eta_j, \tag{24}$$

where $g_{\text{ref},j}$ and $g_{\text{mis},j}$ denote the individual token summands of G_{ref} and G_{mis}. In the ideal preserve setting, if the residual prediction also fits the target so that $\eta_j = 0$, both contributions vanish. In the ideal edit setting, their sum is $-2J_{r,j}^{\mathsf{T}}(x_j + \eta_j) = -2J_{r,j}^{\mathsf{T}}(q_j - r_j)$: the residual branch receives the target error without direct reference reuse. This can produce a larger gradient while the edit is still being learned, provided the target error is nontrivial and the residual Jacobian responds to it.

For the regional interpretation, let $\mathcal{E}$ and $\mathcal{P}$ denote the edited and preserved token sets, respectively, and let $g_j = g_{\text{ref},j} + g_{\text{mis},j}$ denote the total gradient summand for token j.

This distinction is especially relevant to localized edits. When $|\mathcal{E}| \ll |\mathcal{P}|$, most tokens can operate near the preserve endpoint, where reference reuse removes the reference-related gradient, while the smaller edited region operates near the edit endpoint and retains the residual target error. The edited tokens can therefore receive a larger gradient density and a larger share of the update, which is the intended benefit for small editing tasks. If the edit is global, by contrast, $\mathcal{E}$ covers most or all tokens and the routing coefficients can be small across the image. The selective contrast between preserve and edit tokens then largely disappears: the method does not provide the same gradient-concentration advantage, but it also does not suppress the useful training signal because nearly every token requires an update.

The interpretation remains conditional. These endpoint settings are idealized routing states, not spatial labels, and the learned α_j is not an explicit edit mask. Nor does $\alpha_j = 0$ by itself guarantee a large gradient, since the residual may already fit the target and token contributions may cancel. The endpoint argument is token-wise and does not assume any particular relative area of the edited and preserved regions. For an aggregate comparison, define $G_{\mathcal{E}} = \sum_{j\in\mathcal{E}} g_j$ and $G_{\mathcal{P}} = \sum_{j\in\mathcal{P}} g_j$.

Table 4: Semantic task families represented in the supplemental training collection. A family may combine examples from multiple source datasets, and a source dataset may contribute to more than one family.

Task family	Editing operation
Action and pose	Change the action or pose of a depicted subject.
Camera and viewpoint	Change the apparent viewpoint or camera framing.
Chart editing	Modify requested content or appearance in a chart.
Image enhancement	Improve requested aspects of image quality.
Object extraction	Extract a specified object from the source image.
In-image text translation	Translate visible text and render the translated text in the image.
Portrait beautification	Apply requested appearance refinements to portraits.
Spatial relations	Change the relative arrangement of scene elements.
Size adjustment	Change the size of a specified object.
Scene text editing	Modify visible text content or typography.

Region size enters only in this aggregation: if the edited region contains fewer tokens, its average per-token gradient must be correspondingly larger for $\|G_{\mathcal{E}}\|_2$ to dominate $\|G_{\mathcal{P}}\|_2$. Thus, the analysis motivates why edited regions can contribute more strongly during learning, but does not guarantee this inequality for every sample or model state.

C Experimental Setup and Implementation Details

This section provides details on the training data composition, evaluation benchmarks and metrics, compared methods, implementation settings, and student architecture of SPURFLOW.

C.1 Training Data Composition

Base corpus and motivation. We train SPURFLOW on a mixed corpus comprising 257,602 paired examples from Pico-Banana-400K (Qian et al., 2026) and additional editing pairs extracted from publicly available datasets. Pico-Banana-400K defines 35 editing operations and provides the main general-purpose component of our training corpus. However, its task distribution does not directly match the range of editing capabilities covered by GEditBench v2 (Jiang et al., 2026). We therefore add supplemental pairs to increase coverage of geometric, human-centric, text-centric, and other specialized editing operations. Several of these categories overlap with operations already present in Pico-Banana-400K; the supplemental data increases their diversity and representation rather than introducing entirely unseen task types.

Supplemental editing categories. The supplemental pairs cover action and pose changes, camera and viewpoint changes, chart editing, image enhancement, object extraction, in-image text translation, portrait beautification, spatial-relation changes, object-size adjustment, and scene-text editing. Table 4 summarizes these semantic task families. We use approximately 10,000 reference–target pairs for each selected supplemental type.

Sources and example format. Each training example is represented as a reference image, an editing instruction, and a target image, written as $(I_{\mathrm{ref}}, c, I_{\mathrm{tgt}})$. We extract the supplemental examples from task-relevant subsets of AnyEdit (Yu et al., 2025), ScaleEdit-12M (Chen et al., 2026a), UnicEdit-10M (Ye et al., 2026), ChartM^3-Train (Yang et al., 2025), and PPR10K (Liang et al., 2021). AnyEdit, ScaleEdit-12M, and UnicEdit-10M provide broad coverage of action, pose, geometric, object, and text edits. ChartM^3-Train contributes chart-editing examples, while PPR10K contributes portrait-retouching examples. We map the selected source records to the common triplet format and organize them into the semantic families listed in Table 4.

The SFT control in Table 2 also uses Pico-Banana-400K and the same set of supplemental editing categories, rather than Pico-Banana-400K alone.

C.2 Benchmarks and Evaluation Metrics

We evaluate SpurFlow on ImgEdit-Bench Ye et al. (2025) and GEditBench v2 Jiang et al. (2026).

For ImgEdit-Bench, we follow its official evaluation protocol, which scores instruction adherence, editing quality, and detail preservation using task-specific rubrics on a 1–5 scale. We report category-wise scores and the overall average.

For GEditBench v2, we use its evaluation set while following the point-wise evaluation protocol adopted by the original GEdit-Bench Liu et al. (2025), which is based on VIEScore Ku et al. (2024). We report Semantic Consistency (SC), Perceptual Quality (PQ), and Overall Score (O) on a 0–10 scale. We use GPT-4o as the evaluator by default and GPT-4.1 for the Qwen-Image-Edit-2511 comparisons to match the evaluation setting of TwinFlow Cheng et al. (2026). Within each backbone comparison, all methods use the same evaluator and evaluation protocol. Higher scores indicate better performance.

C.3 Baselines

We compare with representative distillation methods, including ArcFlow Yang et al. (2026), SenseFlow Ge et al. (2026), DMD2 Yin et al. (2024a), and UniPic 3.0 Wei et al. (2026). We adapt these methods to instruction-guided image editing using the native reference-image and instruction conditioning pathways of the pretrained editing backbones. Detailed adaptation procedures and baseline-specific configurations are provided in Appendix D.

We further compare with TwinFlow Cheng et al. (2026) and publicly released 4-step editing models, including Qwen-Image-Edit-2511-Lightning LightX2V (2025), TeleStyleV2-QIE-2511-DMD Zhang et al. (2026), and the officially distilled FLUX.2-Klein-9B Labs (2026). The original multi-step teacher models are also included to assess the performance retained after distillation.

C.4 Implementation Details

We distill FLUX.1-Kontext-dev, Qwen-Image-Edit-2511, and FLUX.2-Klein-base-9B into 2-NFE students using paired reference images, editing instructions, and target edits from the mixed training corpus described in Section C.1.

Across all three backbones, we freeze the teacher and the pretrained student DiT blocks, while optimizing rank-256 LoRA adapters, output normalization, residual prediction heads with $K = 16$ mixture components, and an additional token-wise α prediction head.

For edit-aware adversarial learning, we use a frozen DINOv3 ViT-L encoder with trainable source-conditioned discriminator heads. Adversarial supervision combines global views, random local crops, and localized crops guided by the learned α coefficients.

We use a common training framework across the three backbones, while retaining model-specific choices for image preprocessing, guidance, and α activation. Table 5 summarizes the hardware, optimization settings, initialization, training budgets, and adversarial configurations.

C.5 SpurFlow Architecture

Figure 6(a) shows the student backbone, which retains the frozen pretrained DiT blocks and incorporates trainable LoRA adapters. The output normalization and prediction heads are also trainable. The shared hidden representation feeds both the reference-conditioned residual predictor and an additional α_θ head. The residual branch retains the $K = 16$ multi-mode parameterization in Eq. 5 to predict r_θ.

Figure 6(b) details the α_θ head, which uses a linear projection to produce four outputs per 2×2 latent patch. A bounded activation maps these outputs to reference coefficients, allowing reference reuse to vary across spatial positions. Together with the fixed path noise ϵ, the weighted reference and predicted residual form the edit-aware student velocity used in Eq. 4.

Table 5: Implementation details of SPURFLOW across three teacher backbones.

Hyperparameter	FLUX.1-Kontext-dev	Qwen-Image-Edit-2511	FLUX.2-Klein-9B
		General Settings	
Teacher	FLUX.1-Kontext-dev	Qwen-Image-Edit-2511	FLUX.2-Klein-base-9B
Resolution	FLUX buckets	$\sim$1MP resize	$\sim$1MP resize
Latent size	$16 \times 128 \times 128$	$16 \times 128 \times 128$	$32 \times 128 \times 128$
VAE downsampling		$8\times$	
Dataset		Pico-Banana: 257,602	
GPUs		$8\times$ NVIDIA A100	
Per-GPU batch		1	
Global batch		8	
Precision		BF16	
Parallelism		FSDP	
Optimizer		AdamW	
AdamW betas		$(0.9, 0.95)$	
Weight decay		0	
Learning rate		1×10^{-4}	
Schedule		100-iter linear warm-up; constant thereafter	
Clip norm		50 (from iteration 100)	
EMA		Karras $\gamma = 7.0$; start 100, interval 1	
		Warm-up	
Trainable modules		LoRA, output heads, and output normalization	
LoRA		Rank 256; alpha 256; dropout 0.05	
α **channels**		4 per 2×2 patch	
α **activation**	Sigmoid	Softsign-01	Softsign-01
α **init.**		Zero (initial $\alpha_\theta = 0.5$)	
Residual init.		Kaiming He et al. (2015)	
Teacher guidance	Embedded 3.5	CFG 4.0	CFG 4.0
Student guidance	Embedded 3.5	1.0 (cond-only)	1.0 (cond-only)
GM dropout		0.1	
Flow loss		MSE; constant rescaling, scale 30	
Time sampling	shift $\gamma = 3.2$	Dynamic shifting	shift $\gamma = 3.2$
Iterations N	20K	25K	20K
		SPURFLOW Training	
Discriminator backbone		Frozen DINOv3 ViT-L/16; layer 23	
Discriminator head		3 Conv–GroupNorm–SiLU blocks; LayerNorm + Linear	
Discriminator channels		2048	
Discriminator groups		32 (GroupNorm)	
Discriminator optimizer		AdamW	
Discriminator betas		$(0.0, 0.95)$	
Discriminator weight decay		0.01	
Discriminator learning rate		5×10^{-5}	
GAN loss		Logistic (softplus)	
GAN weight		$\lambda_{\text{adv}} = 0.05$	
Update ratio		1:1 (discriminator first)	
Crop size		224 / 224 (global / local)	
Local crops		α-mask or random; $p_{\text{disable}} = 0.3$	
Crop weights		1 / 1 / 1 (global / random / α-mask)	
Iterations N	5K	8K	5K

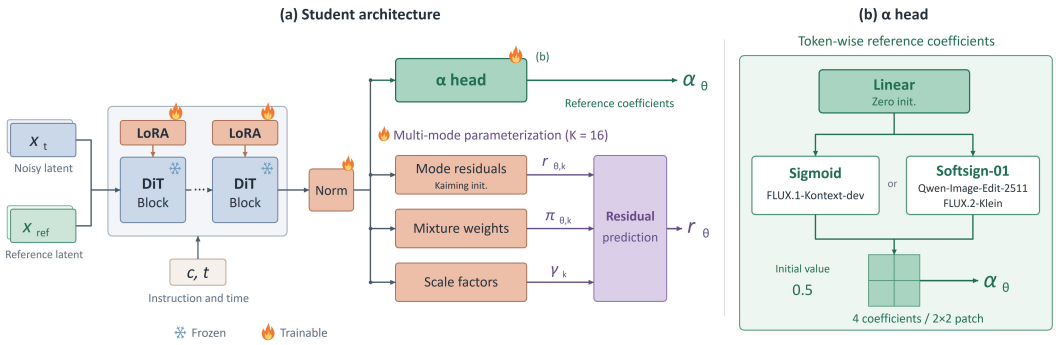

Figure 6: **Student architecture of SPURFLOW.** (a) Frozen pretrained DiT blocks with trainable LoRA adapters, output normalization, and heads for reference coefficients and residual prediction. (b) The additional α_θ head predicts four reference coefficients per 2×2 latent patch, using sigmoid for FLUX.1-Kontext-dev and Softsign-01 for Qwen-Image-Edit-2511 and FLUX.2-Klein.

D BASELINE IMPLEMENTATION DETAILS

ArcFlow, SenseFlow, UniPic 3.0, and DMD2 were developed for text-to-image generation. We adapt them to instruction-guided image editing using pretrained editing backbones, with reference images and editing instructions supplied through their native conditioning pathways. Paired edited targets provide the clean-data and adversarial supervision required by each method. We preserve the original objectives and optimization structures while adjusting the configurations for editing. Each reproduced baseline is trained for 30K iterations.

Below, we detail the FLUX.1-Kontext-dev implementations. The same conditioning and paired-data adaptations are applied to the baselines evaluated on Qwen-Image-Edit-2511 and FLUX.2-Klein-base-9B.

ArcFlow. We retain interval-based velocity matching and the Gaussian-mixture parameterization with $K = 16$ components. Training latents are constructed from paired edited targets, and the teacher and student receive the same reference image and editing instruction. The teacher and pretrained student blocks remain frozen. We optimize rank-256 LoRA adapters, mixture prediction heads, and output normalization.

Training uses AdamW8bit with learning rate 10^{-4}, betas $(0.9, 0.95)$, and zero weight decay. We use one sample per GPU on eight GPUs, giving a global batch size of eight. Teacher and student use embedded guidance 3.5, with a time shift of 3.2. We retain Karras EMA with $\gamma = 7$, starting at iteration 100 and updating every iteration.

SenseFlow. We condition the student, frozen real-score model, and trainable fake-score model on the same reference image and editing instruction. Distribution matching therefore operates under the requested editing conditions. The adversarial branch uses paired target edits as real examples and student-generated edits as fake examples. We retain DMD, ISG, IDA, and the VFM adversarial objective, fully optimizing the student and fake-score models together with the DINOv2-based discriminator heads.

We use AdamW8bit with learning rate 10^{-5}, betas $(0.9, 0.999)$, zero weight decay, and a global batch size of eight. The student uses embedded guidance 3.5. Real-score and fake-score CFG scales are 4.0 and 1.0, respectively. The student is updated once every five iterations. The time-weighted hinge adversarial term has weight 2.0, and the IDA averaging weight is 0.97.

UniPic 3.0. We retain consistency pretraining followed by DMD and incorporate reference-image and instruction conditioning into both stages. Consistency training uses paired edited targets and retains finite-difference tangent estimation with step size 5×10^{-3}, tangent normalization, and the MSE and cosine objectives. The resulting EMA checkpoint initializes the DMD stage, whose fake-

score model receives the same editing conditions as the student. The full student transformer is optimized in both stages, and the fake-score model is trained during DMD.

The student learning rate is 10^{-6} during consistency training and 2×10^{-6} during DMD. The DMD fake-score learning rate is 4×10^{-7}. Both stages use betas $(0.9, 0.95)$ and student EMA with rate 0.95. With one sample per GPU on eight GPUs, gradient accumulation is four during consistency training and one during DMD, giving effective global batch sizes of 32 and eight, respectively. Embedded guidance is 3.5, with CFG 1.75 for consistency training and teacher CFG 6.0 for DMD. During DMD, the student is updated once every five iterations and the fake-score model on the remaining iterations.

DMD2. We instantiate the student, frozen teacher, and trainable fake-score network with the pretrained editing backbone. All three networks receive the same reference image and editing instruction. The ImageDiT adversarial branch uses paired target edits and student-generated edits as real and fake examples, with matched timesteps and noise. We retain the distribution-matching and adversarial objectives and fully optimize the student and fake-score networks together with the discriminator. Teacher and student use embedded guidance 3.5, with additional CFG disabled.

The student, fake-score model, and discriminator use separate AdamW optimizers, each with learning rate 10^{-5}, betas $(0.9, 0.999)$, weight decay 0.01, and no learning-rate warm-up. The global batch size is eight. The student is updated once every five iterations, and the generator adversarial loss has weight 0.03. Training timesteps are sampled uniformly from $[0.001, 0.999]$.

Additional Editing Baselines. We also include TwinFlow in the Qwen-Image-Edit-2511 comparison. For Qwen-Image-Edit-2511-Lightning, TeleStyleV2-QIE-2511-DMD, and FLUX.2-Klein-9B, we directly evaluate the released checkpoints using their provided inference pipelines, guidance settings, and image preprocessing, without additional training. Their inference NFE is specified in Table 1. Within each backbone comparison, all methods are evaluated under the same benchmark protocol as SPURFLOW.

E EDIT-REGION LOCALIZATION EVALUATION

E.1 EVALUATION SUBSET

We use the `PIE-Bench_unique_mask` subset of PIE-Bench Ju et al. (2024), which initially contains 529 samples. We exclude all 22 object-addition examples because the placement of newly added objects is often not uniquely determined, and multiple valid placements may disagree with a single annotated mask. The resulting evaluation subset contains 507 samples across eight categories: random editing (118), object replacement (79), object deletion (77), attribute content (40), attribute pose (40), attribute color (40), attribute material (38), and background change (75). The subset is fixed before evaluation, and all methods are evaluated on the same source images and editing instructions. The corresponding human-annotated masks are used as ground truth for IoU computation.

E.2 COMPARED METHODS AND SETTINGS

SPURFLOW. We use the FLUX.1-Kontext-dev-based 2-NFE student with sigmoid reference coefficients, evaluated at the `iter_11000` checkpoint. Localization uses the continuous first-step α map and the original `step1_mask` processing pipeline. We correct the 2×2 positional bias, apply a 5×5 median filter, and normalize each map using its 2nd and 98th percentiles. We then apply contrast enhancement with strength 40, bicubic upsampling, and Gaussian smoothing with a blur radius of 10 pixels. The processed map is binarized at 0.5, with lower coefficient values assigned to editing regions. White pixels in the resulting binary mask indicate predicted editing regions.

RegionE. We reproduce the adaptive region partition module of RegionE Chen et al. (2026c) using pretrained FLUX.1-Kontext-dev with a 28-step schedule, guidance 2.5, and 1024×1024 inputs. Only the first six warm-up denoising steps are executed. A clean endpoint latent is estimated through velocity extrapolation and compared with the encoded source using token-wise cosine similarity. Tokens with similarity at or below 0.93 are labeled as editing regions.

Table 6: GPT-5.5 evaluation of predicted edit masks on the 507-sample PIE-Bench subset. All metrics range from 0 to 10, with higher scores indicating better performance. Semantic alignment is the primary GPT-based metric.

Method	Sem. Align.↑	Coverage↑	Precision↑	Compactness↑
RegionE	5.519	7.057	**4.787**	3.400
RWTE	5.083	6.071	4.473	3.258
SPURFLOW	**5.927**	**7.187**	4.673	**4.030**

RWTE. We use a local reproduction of the task-aware localization strategy of RWTE He et al. (2026), with the same pretrained backbone, sampling schedule, guidance, and input resolution as RegionE. Attention and latent features are extracted from denoising steps 2–5 and double-stream transformer blocks 10, 14, and 18. The top 10% and bottom 40% of attention scores initialize editing and preservation centroids, corresponding to attention quantiles of 0.90 and 0.40, respectively. Feature clustering combines attention and feature cues with equal weights of 0.5, followed by task-aware fusion of source and target masks.

Both baselines use random seed 110. Their predicted masks undergo 3×3 cross-shaped erosion followed by 5×5 square dilation.

E.3 MASK IOU

Predicted binary masks are aligned with the 512×512 ground-truth masks using nearest-neighbor resizing when necessary. For a predicted edit region $\widehat{M}$ and ground-truth edit region M_{gt}, IoU is defined as

$$\mathrm{IoU}(\widehat{M}, M_{\mathrm{gt}}) = \frac{|\widehat{M} \cap M_{\mathrm{gt}}|}{|\widehat{M} \cup M_{\mathrm{gt}}|}. \tag{25}$$

Category-wise scores average per-image IoU within each category, while the overall score averages per-image IoU across all 507 samples.

E.4 GPT-BASED SEMANTIC EVALUATION

Evaluation protocol. To complement annotation-based IoU, we evaluate whether predicted masks identify semantically appropriate edit regions using the API model identifier `gpt-5.5`. Each request contains only the original PIE-Bench image, the editing instruction, the predicted binary mask, and a visualization of that mask overlaid on the original image. White pixels indicate regions predicted to require editing. No teacher-generated images, student-generated images, other target images, or human-annotated ground-truth masks are provided to the judge. The judge independently determines which regions a competent human editor would need to modify, based on the source image and instruction, and assesses the predicted mask accordingly.

Input images are resized to a long-edge resolution of 768 pixels. Each sample–method pair is evaluated with one API call, with up to three retries following API failures. All three methods successfully complete evaluation on all 507 samples, with no unresolved failures.

Metrics and aggregation. The judge returns four scores on a 0–10 scale: *coverage*, measuring whether all necessary edit regions are included; *precision*, measuring whether regions that should remain unchanged are excluded; *compactness*, measuring whether the mask selects the smallest coherent region sufficient for the edit; and *semantic alignment*, measuring whether the correct object, part, or background is selected. Semantic alignment is the primary GPT-based metric reported in the main paper. Each metric is averaged over the same 507 samples for each method.

Table 6 reports all four metrics. SPURFLOW achieves the highest semantic alignment, coverage, and compactness scores, while RegionE obtains the highest precision.

Evaluation prompt. The following system prompt is used for all methods.

System prompt.

You are evaluating intended edit-region localization.

You will receive the original image, an editing instruction, a predicted binary mask, and the same mask overlaid on the original image. White means "edit this region".

Do not infer the answer from a generated image. Independently determine which region a competent human editor would need to modify to satisfy the instruction. Judge whether the mask identifies the smallest sufficient semantic region.

Task-specific guidance:

- Object replacement or attribute change: select the affected object or relevant object part, not unrelated surroundings.
- Object deletion: include the object and only the minimal surrounding area needed for inpainting.
- Pose change: include all articulated parts that must move, but avoid unrelated background.
- Background change: include the background while preserving foreground subjects.
- A near-full-image mask is valid only for a truly global transformation.

Evaluate:

- `coverage`: all regions that must be edited are included;
- `precision`: regions that should remain unchanged are excluded;
- `compactness`: the mask is the smallest coherent region sufficient for the edit;
- `semantic_alignment`: the correct object, part, or background is selected.

Do not judge image quality, generation quality, realism, or instruction execution. Do not reward a mask merely for being large.

Return a single JSON object and nothing else:

```
{"coverage": <0-10>, "precision": <0-10>, "compactness": <0-10>,
 "semantic_alignment": <0-10>, "reason": "<one concise sentence>"}
```

F Additional Qualitative Results

We provide additional qualitative comparisons on ImgEdit-Bench and GEdit-Bench v2 to complement the quantitative results in the main paper. These examples further illustrate the editing fidelity, visual quality, and reference preservation of SpurFlow across diverse editing instructions.

F.1 Additional Results on ImgEdit-Bench

Fig. 7 presents additional qualitative comparisons on ImgEdit-Bench, covering action, object addition and removal, attribute adjustment, background editing, composition, extraction, replacement, and style transfer. With only two sampling steps, SpurFlow achieves visual quality broadly comparable to its multi-step teachers and, in some cases, follows the editing instructions more faithfully, as illustrated by the object removal example.

F.2 Additional Results on GEditBench v2

Fig. 8 presents additional qualitative comparisons on GEditBench v2, which includes diverse and challenging real-world editing instructions. Across different editing scenarios, SpurFlow produces high-quality results with strong instruction adherence while preserving source content that should remain unchanged. Despite using only two sampling steps, its results are visually competitive with multi-step teachers and strong few-step baselines, demonstrating the effectiveness and generalization of our edit-aware distillation framework.

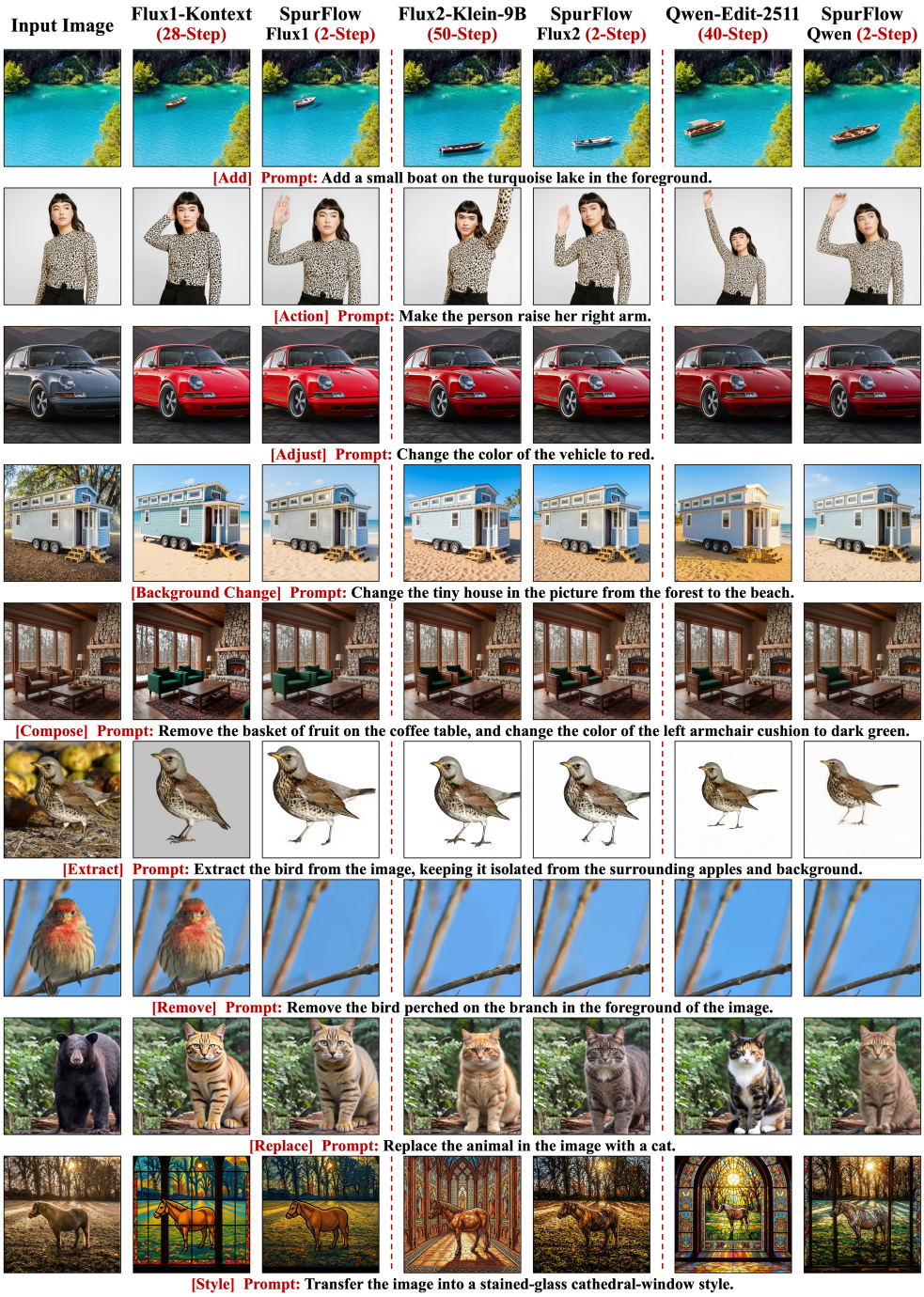

Figure 7: **Additional qualitative results on ImgEdit-Bench.** Comparisons across diverse image editing categories, highlighting instruction adherence, visual quality, and source-content preservation.

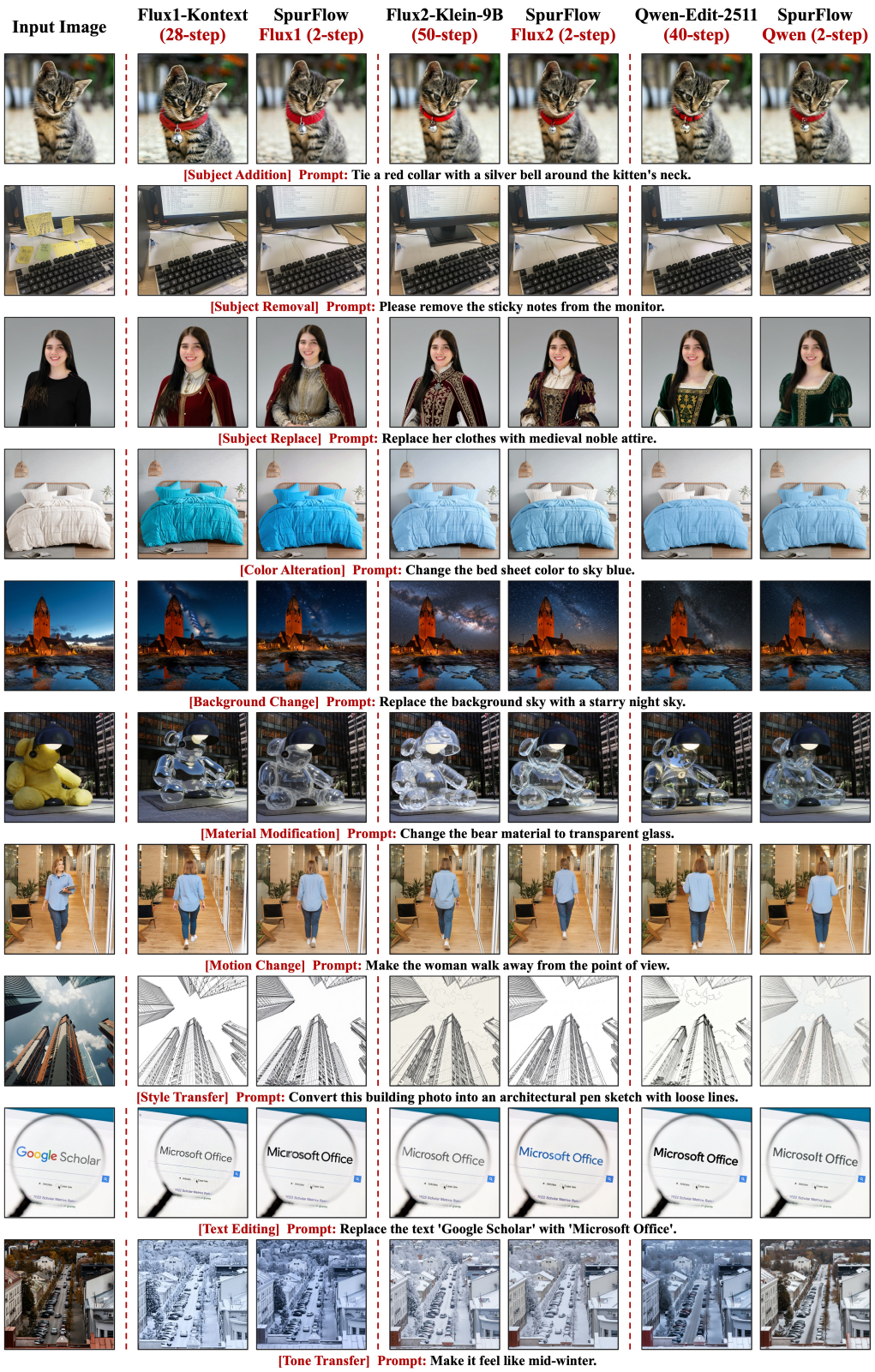

Figure 8: **Additional qualitative results on GEditBench v2.** SPURFLOW achieves strong instruction adherence and source-content preservation across diverse real-world editing requests using only two sampling steps.