SpatialFlow-GRPO: Where Spatial Credit Drives Image Editing

Yankai Yang[*1,2] **Yancheng Long**[*1,2] **Wei Chen**[2] **Xingyu Lu**[2] **Hongyang Wei**[2]
Bin Wen[†✉2] **Fan Yang**[2] **Tingting Gao**[2] **Han Li**[2] **Shuo Yang**[✉1]

Abstract

Recent online reinforcement learning has substantially improved image editing quality. However, existing Flow-GRPO-style methods usually rely on a single whole-image reward, which makes fine-grained editing optimization difficult. We observe that a key obstacle in image editing is this spatial uniformity assumption: a whole-image reward cannot distinguish how different spatial regions contribute to image quality. To address this issue, we propose SpatialFlow-GRPO, a training framework that introduces spatially fine-grained reward feedback. The framework converts region-aware rewards into semantic-region-level optimization signals and aligns region advantages with the corresponding latent positions during policy updates. We also train a region-aware reward model, SFReward, construct SFReward-14K with region-annotated editing samples, and introduce MultiEdit-Bench to evaluate multi-region editing ability. On OmniGen2 and FLUX.2-klein-4B, SpatialFlow-GRPO outperforms Flow-GRPO with the same reward model on GEdit-Bench, ImgEdit-Bench, and MultiEditBench. The results show that SpatialFlow-GRPO converts local feedback into spatially aligned update signals and improves editing quality.

1 Introduction

Diffusion models have made rapid progress in image generation and editing [15, 41, 35, 8]. Image editing has therefore become a core ability of visual generative models [5, 32, 6, 14, 10]. Unlike open-ended text-to-image generation, editing quality is often spatially non-uniform. Different regions in the same output may show different degrees of completion, failure, or quality degradation. Thus, post-training for image editing needs more than a global preference signal. It also needs fine-grained feedback that can indicate quality differences across spatial regions.

Recent work has introduced online reinforcement learning (RL) into diffusion model post-training. Reward models provide preference, aesthetic, or editing-quality feedback to improve generation quality and instruction alignment [2, 9, 13]. However, methods such as Flow-GRPO [27] and DanceGRPO [50] usually treat an edited image as a single outcome. They use one whole-image reward to compute a sample-level advantage and apply the same advantage to all latent positions. This design implies a **spatial uniformity assumption**: different regions in the image share the same credit signal. This assumption limits fine-grained optimization in editing tasks, because a single whole-image score cannot determine which target region accounts for editing success or failure, or whether the change reflects preservation quality in unedited regions.

Figure 1 shows a multi-region editing example: the instruction modifies the skirt, adds a dog, replaces a wall symbol with the letter V, and removes a person from the doorway. Flow-GRPO receives

[*]Equal contribution. [†]Project leader. [✉]Corresponding authors.
[1]Harbin Institute of Technology, Shenzhen. [2]Kuaishou Technology.
Correspondence to: Shuo Yang <shuoyang@hit.edu.cn>, Bin Wen <wenbin@kuaishou.com>.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

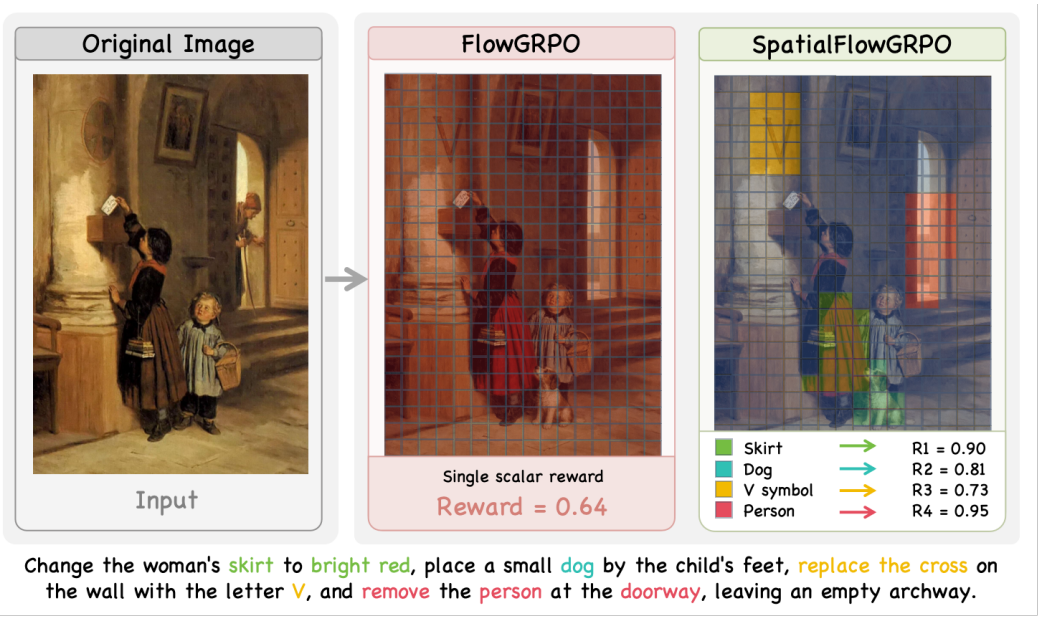

Figure 1: **Motivation.** In multi-region editing, different regions may have different quality outcomes. Flow-GRPO collapses them into one scalar reward, while SpatialFlow-GRPO attaches reward feedback to semantic regions and enables spatially localized credit assignment.

only one scalar reward for the whole result, so local successes and failures are collapsed into the same feedback. Region-level feedback can instead assign separate scores to the affected semantic regions, making the update signal spatially localizable. We call this the **spatial credit assignment dilemma**: quality differences in image editing have spatial structure, but the RL supervision signal and update rule lack matching spatial resolution. This problem is closely related to credit assignment and reward design in reinforcement learning [37].

Based on this observation, we propose SpatialFlow-GRPO, which extends the sample-level relative optimization of Flow-GRPO to spatially structured region-level optimization. The framework uses SFReward to produce quality feedback tied to semantic regions. It compares semantically corresponding regions among candidates sampled for the same instruction, producing localized region advantages. SpatialFlow-GRPO further preserves this spatial correspondence in the policy objective. Region advantages are applied to the corresponding latent positions, and region-consistent policy ratios with region-weighted aggregation reduce the dilution of local signals by whole-image averaging. As a result, RL post-training no longer updates the model only according to whole-image quality. It can use region-level quality differences for more precise spatial credit assignment.

The main contributions are summarized as follows:

1. **Identifying the spatial uniformity limitation.** We identify that whole-image rewards in Flow-GRPO-style methods force different spatial regions to share the same credit signal. This is a key factor that limits fine-grained optimization for image editing.

2. **Region-level RL training framework.** We propose SpatialFlow-GRPO, which integrates region-aware rewards, semantic-group advantage estimation, and a latent-region-aligned policy objective into online RL post-training. This allows local feedback to serve as localized training signals for model updates.

3. **Region reward model, data, and benchmark.** We train the region-aware reward model SFReward, build SFReward-14K for region reward learning, and introduce MultiEditBench for diagnosing fine-grained editing ability. On OmniGen2 and FLUX.2-klein-4B, SpatialFlow-GRPO outperforms Flow-GRPO with the same reward model on GEdit-Bench, ImgEdit-Bench, and MultiEditBench.

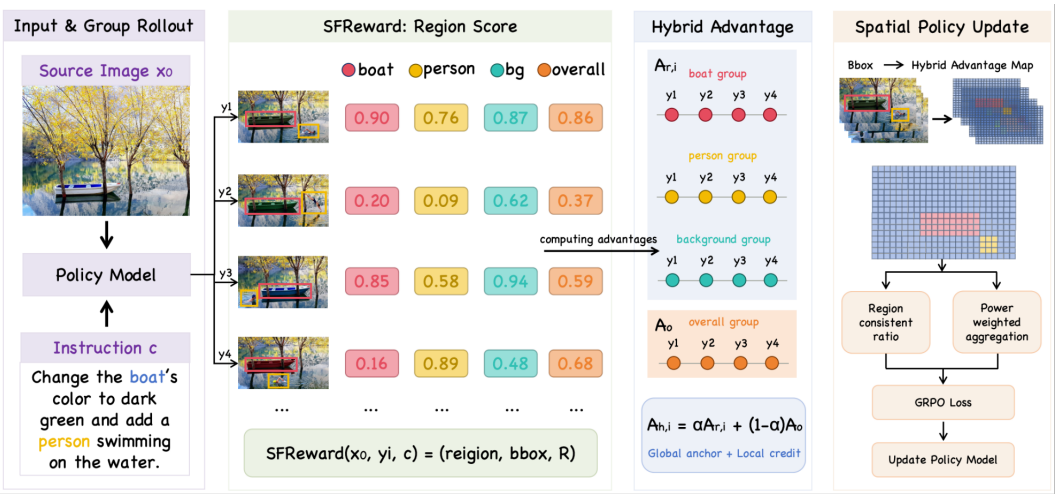

Figure 2: **Overview of SpatialFlow-GRPO.** The policy samples a group of edited images for each instruction, and SFReward returns region boxes, semantic labels, and scores. Region and global scores are converted into hybrid advantages, mapped back to latent regions, and optimized through region-consistent ratios and power-weighted aggregation.

2 Related work

Image editing. Image editing modifies a source image according to text while preserving naturalness and semantic consistency. Prior methods use noise perturbation, masks, or attention control [32, 6, 14], and instruction-based data and models further broaden editing ability [5, 54, 10]. Recent systems scale data, model interfaces, and task coverage [55, 20, 21, 46, 28, 40, 39, 44]. We study how to post-train such editors with spatially fine-grained RL feedback.

RL post-training for diffusion models. RL post-training directly optimizes diffusion policies with reward or preference signals [2, 9, 43], including image-editing variants [25, 24]. DanceGRPO [50] and Flow-GRPO [27] extend GRPO to visual generation and flow matching, while ImageDoctor [13] studies diagnosis-oriented visual evaluation. These methods improve alignment but usually compress each edited image into one sample-level reward, which can hide local quality variation.

Fine-grained credit assignment and region-aware rewards. Fine-grained credit assignment improves RL efficiency and stability. GRPO-style reasoning systems show the value of relative optimization [38, 12, 53], and B2-DiffuRL densifies sparse diffusion rewards over timesteps [18]; however, image editing also requires spatial credit. Visual reward models have become stronger [49, 22, 48, 30, 47, 51, 29]: SpatialReward grounds editing judgments in predicted edit regions, and SpatialScore [42] models spatial relations for text-to-image generation, but both return image-level scores that enter RL as sample-level advantages. Our contribution is complementary: we train an editing-oriented region evaluator, SFReward, and convert its per-region scores into comparable region advantages while preserving spatial correspondence during policy updates.

Evaluation benchmarks and fine-grained editing diagnosis. Editing evaluation is also moving from overall scores toward detailed diagnosis. Existing benchmarks cover real user requests, general editing, multi-attribute/object editing, data quality, automatic evaluation, interpretable scoring, and spatial preservation [28, 52, 19, 21, 31, 23, 29]. They measure overall quality well, but rarely isolate whether a model handles spatially distinct goals as editing complexity increases. We introduce MultiEditBench to evaluate this fine-grained ability under multi-region edits.

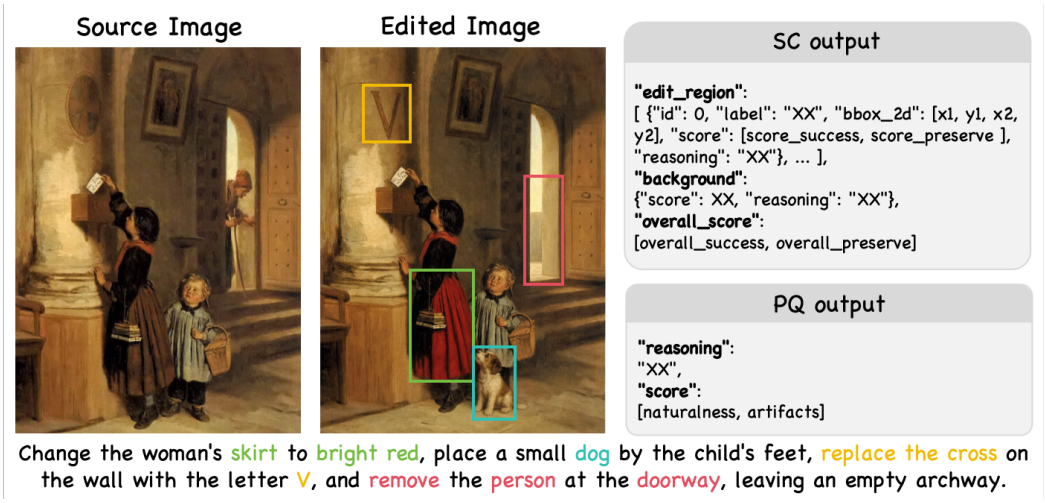

Figure 3: **Structured output format of SFReward.**

3 Method

The goal of SpatialFlow-GRPO is to turn whole-image relative optimization in GRPO into spatial credit assignment over semantic regions. Given an editing instruction, a source image, and G candidate outputs sampled from the current policy, SFReward predicts semantic regions to be scored, their bounding boxes, and rewards $\{R_{i,r}\}$ for each output. SpatialFlow-GRPO compares semantically corresponding regions under the same instruction and combines region-level advantages with a global quality anchor. The resulting hybrid advantages are aligned with the corresponding latent positions and used in the policy objective. In this way, the reward no longer affects the whole image only at the sample level. Instead, it can guide model updates at the region level. The overall framework is shown in Figure 2.

3.1 Problem setup and structured region reward

We follow the online training paradigm of Flow-GRPO [27]. Given an editing instruction c and a source image, the policy model generates G edited images in each sampling round, together with their reverse diffusion trajectories $\{x_t^i\}_{t=0}^T$ [15, 26]. Standard Flow-GRPO uses a sample-level reward R_i to compute the within-group advantage:

$$\hat{A}_i = \frac{R_i - \mu_R}{\sigma_R + \epsilon},$$

and optimizes a PPO/GRPO-style clipped surrogate over all timesteps [36]:

$$J_{\text{Flow-GRPO}}(\theta) = \frac{1}{GT}\sum_{i=1}^{G}\sum_{t=0}^{T-1}\min\Big(r_{i,t}(\theta)\hat{A}_i,\ \text{clip}(r_{i,t}(\theta), 1-\epsilon_-, 1+\epsilon_+)\hat{A}_i\Big) - \lambda_{\text{kl}}\mathcal{L}_{\text{kl}}, \quad (1)$$

where

$$r_{i,t}(\theta) = \frac{p_\theta(x_{t-1}^i \mid x_t^i, c)}{p_{\theta_{\text{old}}}(x_{t-1}^i \mid x_t^i, c)}.$$

This objective fits global preference feedback but loses local editing success and failure in multi-region editing. Our Flow-GRPO baseline uses the sample-level score from the same SFReward as R_i, while SpatialFlow-GRPO uses region scores to construct region-level advantages and objectives.

SFReward outputs two types of structured scores for each edited result. Semantic consistency (SC) evaluates whether the edit succeeds and whether it avoids over-editing. Perceptual quality (PQ) evaluates image naturalness and artifacts. The output format is shown in Figure 3. Let $\mathbf{o}_i$ denote the `overall_score` from SC, and let $\mathbf{q}_i$ denote the `score` from PQ. The Flow-GRPO baseline and the reward-model evaluation use a VIEScore-style aggregation [23] to combine global SC and PQ

scores into a sample-level reward:

$$R_i^g = \frac{\sqrt{\min(\mathbf{o}_i) \cdot \min(\mathbf{q}_i)}}{C}, \tag{2}$$

where C is a fixed normalization constant.

Unlike the sample-level reward in Equation (2), SpatialFlow-GRPO computes a separate reward for each edited region and for the background. SFReward provides semantic-consistency scores for the regions in `edit_region`, a background-preservation score, and image-level perceptual-quality scores. For sample i, let $\mathcal{R}_i = \mathcal{R}_i^{\text{fg}} \cup \{\text{bg}\}$, let $\text{IF}_{i,r}$ denote the semantic-consistency score of foreground region r, let $\text{IF}_{i,\text{bg}}$ denote the background score, and let $\text{AES}_i = \min(\mathbf{q}_i)$ denote the perceptual-quality term. The region reward is

$$R_{i,r} = \frac{\sqrt{\phi(\text{IF}_{i,r}) \cdot \text{AES}_i}}{C}, \qquad \phi(\text{IF}_{i,r}) = \begin{cases} \min(\text{IF}_{i,r}), & r \in \mathcal{R}_i^{\text{fg}}, \\ \text{IF}_{i,\text{bg}}, & r = \text{bg}, \end{cases} \tag{3}$$

where ϕ selects the local semantic-consistency signal for each foreground region and the preservation signal for the background. AES_i is computed from image-level perceptual-quality scores, since naturalness, artifacts, and aesthetic quality are judged from the whole edited image rather than from isolated regions. It therefore serves as a global quality factor shared by all region rewards. SFReward is trained on SFReward-14K, a 14,276-sample region-annotated editing dataset; details are provided in Appendix A.4.

3.2 Semantic-region advantage estimation

Rewards are comparable only within the same instruction and semantic label. SpatialFlow-GRPO therefore normalizes rewards by $(\text{instruction}, \text{label})$. Given instruction p and label l, we define:

$$\mathcal{S}_{p,l} = \{R_{i,r} \mid c_i = p,\ l_{i,r} = l\}. \tag{4}$$

When $|\mathcal{S}_{p,l}| \geq 2$, the region advantage is

$$\hat{A}_{i,r} = \frac{R_{i,r} - \mu_{p,l}}{\sigma_{p,l} + \epsilon}, \tag{5}$$

where $\mu_{p,l}$ and $\sigma_{p,l}$ are the mean and standard deviation within the semantic group. Since grouping is within the same instruction, no predefined global semantic taxonomy is required.

Region advantages capture local quality differences, but pure local scores can weaken composition, identity preservation, and naturalness. We therefore mix each region advantage with the global advantage:

$$\hat{A}_{i,r}^{\text{mix}} = (1-\alpha)\hat{A}_i^g + \alpha\hat{A}_{i,r}, \quad \alpha \in [0,1]. \tag{6}$$

Here, the global advantage $\hat{A}_i^g$ is computed from the sample-level reward R_i^g in Equation (2). Early in training, a small α preserves the global quality constraint. The region signal is then gradually increased using a clipped cosine schedule:

$$\alpha(s) = \alpha_{\min} + \frac{\alpha_{\max} - \alpha_{\min}}{2}\left(1 - \cos(\pi u(s))\right), \tag{7}$$

where $u(s) = \text{clip}((s - s_{\text{start}})/(s_{\text{end}} - s_{\text{start}}), 0, 1)$. Each region box is scaled to latent resolution and mapped to positions $M_{i,r}$; positions in $M_{i,r}$ use $\hat{A}_{i,r}^{\text{mix}}$, while uncovered positions use the background region. Overlapping region boxes and sparse semantic groups are handled by multi-region averaging and a global-advantage fallback, respectively; implementation details are provided in Appendix A.3.

3.3 Region-aligned policy objective

Region advantages describe editing-quality differences of semantic regions relative to the candidates in the same group. To preserve this correspondence during the update, the policy objective should also operate at the region level. If policy ratios and loss aggregation are still computed independently over latent positions, the region-level signal can be diluted by local noise along the diffusion trajectory and by the large area of the background. SpatialFlow-GRPO therefore introduces two region-aligned designs in the policy objective.

First, we aggregate position-level log-ratios within each region:

$$\delta_{i,t,k} = \log p_\theta(x^i_{t-1} \mid x^i_t, c)_k - \log p_{\theta_{\text{old}}}(x^i_{t-1} \mid x^i_t, c)_k, \tag{8}$$

$$\bar{\delta}_{i,t,r} = \frac{1}{|M_{i,r}|} \sum_{k \in M_{i,r}} \delta_{i,t,k}, \qquad s_{i,t,r} = \exp(\bar{\delta}_{i,t,r}). \tag{9}$$

All latent positions in the same region share the region ratio $s_{i,t,r}$, so the measure of policy change has the same granularity as the region advantage.

Second, to prevent whole-image averaging from weakening foreground signals, we average within each region and use power-weighted aggregation across regions:

$$\ell_{i,t,k} = \max\Big(-\hat{A}^{\text{mix}}_{i,k}\, s_{i,t,r(k)},\ -\hat{A}^{\text{mix}}_{i,k}\, \text{clip}(s_{i,t,r(k)}, 1-\epsilon_-, 1+\epsilon_+)\Big), \tag{10}$$

where $r(k)$ is the region that contains position k. The policy loss is:

$$L^{(t)}_{\text{pg}} = \frac{1}{G} \sum_{i=1}^{G} \sum_{r \in \mathcal{R}_i} \tilde{w}_{i,r} \frac{1}{|M_{i,r}|} \sum_{k \in M_{i,r}} \ell_{i,t,k}, \tag{11}$$

where

$$\tilde{w}_{i,r} = \frac{(|M_{i,r}| + \tau)^\beta}{\sum_{r' \in \mathcal{R}_i} (|M_{i,r'}| + \tau)^\beta}, \quad \beta \in [0,1],\ \tau > 0. \tag{12}$$

When $\beta = 1$, aggregation is area-weighted; when $\beta = 0$, regions are weighted equally. We use $\beta = 0.7$ and $\tau = 10^{-2}$ to preserve area information while increasing foreground influence.

The final training objective is

$$\mathcal{L}_{\text{SpatialFlow-GRPO}}(\theta) = \frac{1}{|\mathcal{T}_s|} \sum_{t \in \mathcal{T}_s} \omega_t L^{(t)}_{\text{pg}} \ + \ \lambda_{\text{kl}} \frac{1}{|\mathcal{T}_s|} \sum_{t \in \mathcal{T}_s} L^{(t)}_{\text{kl}}, \tag{13}$$

where $\mathcal{T}_s$ is the sampled timestep set, ω_t follows Flow-GRPO, and $L^{(t)}_{\text{kl}}$ regularizes against the reference policy. Unlike Flow-GRPO, SpatialFlow-GRPO keeps semantic-region granularity in advantages, ratios, and aggregation, preserving spatial attribution during updates.

The full training procedure, including region rewards, semantic advantage estimation, and the region-aligned policy objective, is summarized in Algorithm 1 in Appendix A.2.

4 Experiments

We evaluate the reward model, standard editing performance, multi-region editing ability, and component ablations.

4.1 Experimental setup

We evaluate SpatialFlow-GRPO on **OmniGen2** [46], a Diffusion Transformer [34], and **FLUX.2-klein-4B** [3], a Rectified Flow model [26, 8]. The main baseline is Flow-GRPO [27]. Both methods use the same reward model, training data, base model, and sampling settings; Flow-GRPO uses the sample-level score in Equation (2) as the reward, while SpatialFlow-GRPO uses region scores to construct region advantages and the region-aligned policy objective.

Metrics include three benchmarks: GEdit-Bench [28], ImgEdit-Bench [52], and MultiEditBench. RL post-training uses the 9-category editing subset of EditScore-RL-Data [30] as the source of editing instructions and source images. Training parameters and implementation details are provided in Appendix A.1.

4.2 SFReward evaluation

We compare SFReward with closed- and open-source evaluators on EditReward-Bench [30] and MMRB2 [17]. SFReward is initialized from Qwen3-VL-8B-Instruct [1]; the "w/o Dense Anno." variant uses the same data source but keeps only the original global annotations, without the added region-level annotations. Table 1 shows that dense region supervision improves the evaluator's ability to judge editing quality.

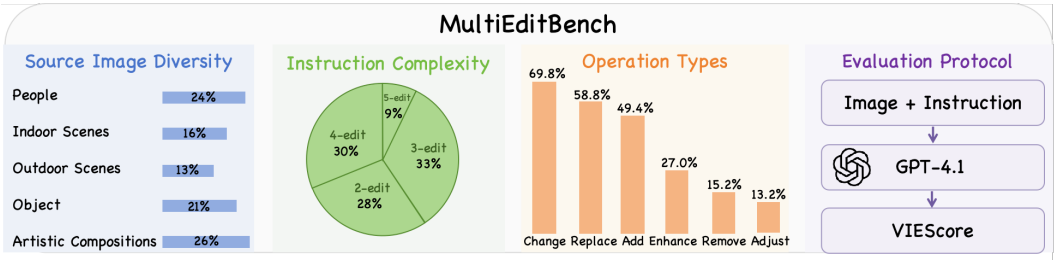

Figure 4: **Composition of MultiEditBench.**

Table 1: **SFReward evaluation results.** PF, Cons., and Ovrl. denote prompt following, consistency, and overall, respectively.

Model	Type	EditReward-Bench			MMRB2		
		PF	Cons.	Ovrl.	Single	Multi	Ovrl.
Closed-source models							
GPT-4.1	Closed	0.673	0.602	0.705	0.547	0.478	0.535
GPT-5	Closed	0.777	0.669	0.755	0.627	0.584	0.619
Gemini-2.5-Pro	Closed	0.703	0.560	0.722	0.545	0.483	0.534
Gemini-3.0-Flash	Closed	0.717	0.662	**0.769**	0.627	0.596	**0.621**
Open-source generative evaluators							
Qwen3-VL-8B	Gen.	0.419	0.243	0.562	0.425	0.393	0.419
EditScore-7B (Avg@4)	Gen.	0.722	0.720	0.727	–	–	–
w/o Dense Anno. (Ours)	Gen.	0.627	0.514	0.699	0.591	0.545	0.583
SFReward (Ours)	Gen.	0.648	0.621	**0.760**	0.629	0.551	**0.615**

4.3 Main results

We further evaluate the effectiveness of SpatialFlow-GRPO as a region-level RL post-training method. Table 2 reports standard benchmark results on two base models. Across both OmniGen2 and FLUX.2-klein-4B, SpatialFlow-GRPO consistently improves over Flow-GRPO with the same reward model.

These gains are consistent with the difference in optimization granularity. If region-level feedback is compressed into a sample-level score, local error sources are averaged during the update. In contrast, SpatialFlow-GRPO preserves region attribution in advantage estimation, policy ratios, and loss aggregation, allowing feedback to be converted into editing improvements more effectively. This trend appears on both Diffusion Transformer and Rectified Flow backbones, supporting the effectiveness of fine-grained credit assignment itself. Figure 5 provides qualitative examples of this behavior, with more results in Appendix A.9.

Since several gains are small, especially on FLUX.2-klein-4B, Table 3 repeats the main comparison with three training seeds. SpatialFlow-GRPO outperforms Flow-GRPO under every seed in all six backbone–benchmark pairs, and the seed standard deviations (0.01–0.05) stay below the corresponding mean gains.

We also train Flow-GRPO on OmniGen2 with SpatialReward [29] as the reward (Table 2). SpatialReward grounds its judgments in predicted edit regions but returns whole-image scores rather than independent per-region scores, so it cannot drive the region-aligned objective of SpatialFlow-GRPO; it is also trained on about 18$\times$ more data than SFReward (260K vs. 14.3K samples). This variant is slightly better on GEdit-Bench (7.32 vs. 7.29) and comparable on ImgEdit-Bench (3.72 vs. 3.74), whereas SpatialFlow-GRPO is clearly better on MultiEditBench (8.11 vs. 7.86; Flow-GRPO with SFReward: 7.52), suggesting that the multi-region gain comes from region-level credit assignment rather than from reward-model scale.

Table 2: **Standard benchmark results.** Overall is the score after post-training, and Δ is the gain over the corresponding base model without RL. Flow-GRPO and SpatialFlow-GRPO use SFReward unless the reward model is given in parentheses. Bold marks the best result for each base model.

Base model	Method	GEdit Overall ↑	Δ GEdit ↑	ImgEdit Overall ↑	Δ ImgEdit ↑
OmniGen2	No RL	6.42	0.00	3.44	0.00
OmniGen2	Flow-GRPO	6.99	+0.57	3.66	+0.22
OmniGen2	Flow-GRPO (SpatialReward)	**7.32**	**+0.90**	3.72	+0.28
OmniGen2	SpatialFlow-GRPO	7.29	+0.87	**3.74**	**+0.30**
FLUX.2-klein-4B	No RL	6.58	0.00	3.42	0.00
FLUX.2-klein-4B	Flow-GRPO	6.73	+0.15	3.50	+0.08
FLUX.2-klein-4B	SpatialFlow-GRPO	**6.80**	**+0.22**	**3.53**	**+0.11**

Table 3: **Seed robustness.** Mean $\pm$ sample standard deviation over three training seeds. Gain is the mean paired improvement of SpatialFlow-GRPO over Flow-GRPO under the same seed (computed from unrounded means); it is positive for every seed in all six comparisons. All other RL results are from single training runs.

Base model	Benchmark	Flow-GRPO	SpatialFlow-GRPO	Gain
OmniGen2	GEdit-Bench	6.99 ± 0.04	7.28 ± 0.03	+0.29
	ImgEdit-Bench	3.63 ± 0.03	3.72 ± 0.03	+0.10
	MultiEditBench	7.53 ± 0.05	8.13 ± 0.02	+0.60
FLUX.2-klein-4B	GEdit-Bench	6.71 ± 0.02	6.80 ± 0.01	+0.10
	ImgEdit-Bench	3.51 ± 0.01	3.56 ± 0.03	+0.05
	MultiEditBench	7.93 ± 0.04	8.03 ± 0.03	+0.10

4.4 MultiEditBench: multi-region editing diagnosis

MultiEditBench evaluates a model's ability to follow complex instructions that require multiple simultaneous edits. It contains 500 manually reviewed samples with 2–5 editing targets, which are grouped into 2-edit, 3-edit, 4-edit, and 5-edit subsets. We use GPT-4.1 with the VIEScore

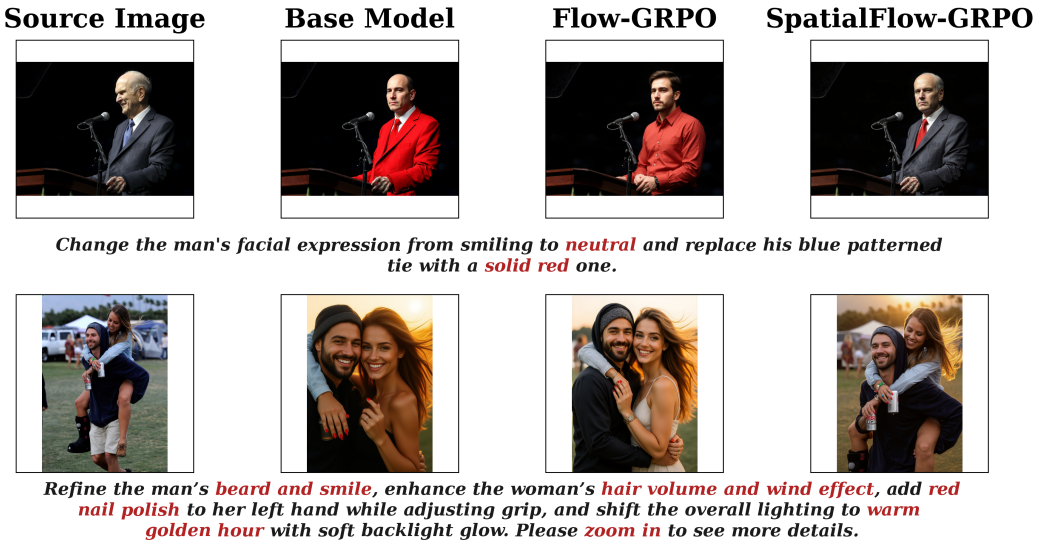

Figure 5: **Qualitative examples of SpatialFlow-GRPO on OmniGen2.** Compared with Flow-GRPO, SpatialFlow-GRPO better preserves source identity and applies multiple requested local edits.

Table 4: **Full MultiEditBench results.** MEB Score is reported by the number of simultaneous editing regions. Avg is the equal-weighted average over the four difficulty subsets. Δ Avg denotes the improvement over the corresponding base model and is omitted for standalone reference editors.

Model	2-edit (n=140)	3-edit (n=165)	4-edit (n=150)	5-edit (n=45)	Avg	Δ Avg
Qwen-Image-Edit-2509 [45]	8.39	8.39	8.33	8.23	8.33	–
OmniGen2 + SpatialFlow-GRPO	8.18	8.19	8.04	8.03	8.11	**+1.70**
FLUX.2-klein-4B + SpatialFlow-GRPO	8.03	8.07	8.14	7.92	8.04	**+0.19**
FLUX.2-klein-4B + Flow-GRPO	7.98	8.06	7.99	7.84	7.97	+0.12
FLUX.2-klein-4B Base	7.84	7.85	7.99	7.73	7.85	–
OmniGen2 + Flow-GRPO	7.52	7.42	7.40	7.72	7.52	+1.11
Step1X-Edit [28]	7.28	7.12	7.12	7.14	7.16	–
FLUX.1-Kontext [4]	6.80	7.04	6.80	7.66	7.07	–
BAGEL-7B-MoT [7]	6.67	6.74	6.50	6.59	6.62	–
OmniGen2 Base	6.80	6.43	6.32	6.09	6.41	–

protocol [23] to judge semantic consistency (SC) and perceptual quality (PQ). Figure 4 summarizes the benchmark composition, and the construction process is described in Appendix A.5.

Table 4 compares SpatialFlow-GRPO with Flow-GRPO, the corresponding base models, and representative open-source editors. Across both OmniGen2 and FLUX.2-klein-4B, SpatialFlow-GRPO gives the strongest result among methods using the same backbone, suggesting that region-level credit assignment improves multi-target editing beyond standard whole-image reward optimization.

4.5 Component and parameter ablations

The ablation study analyzes SpatialFlow-GRPO from two aspects: parameter sensitivity and component contribution.

Table 5: β–α **parameter scan** (OmniGen2, SFReward reward model). Overall is the score, and Δ is the gain over the OmniGen2 base model (GEdit 6.42, ImgEdit 3.44). "w" indicates cosine warmup.

β	α	GEdit Overall ↑	Δ GEdit ↑	ImgEdit Overall ↑	Δ ImgEdit ↑
$\beta = 1$ *(token-weighted, equivalent to global averaging)*					
1	1 (pure region advantage)	6.92	+0.50	3.63	+0.19
1	0.5	7.04	+0.62	3.68	+0.24
1	0.7	6.98	+0.56	3.64	+0.20
1	0.3	7.02	+0.60	3.68	+0.24
1	0.5w	7.07	+0.65	3.67	+0.23
$\beta = 0$ *(equal region weighting)*					
0	1	6.94	+0.52	3.63	+0.19
0	0.5	7.12	+0.70	3.70	+0.26
0	0.5w	7.17	+0.75	3.69	+0.25
0	0.7w	7.01	+0.59	3.66	+0.22
$\beta = 0.7$ *(power-weighted trade-off)*					
0.7	0.5w	**7.29**	**+0.87**	**3.74**	**+0.30**
0.7	0.5	7.21	+0.79	3.72	+0.28
0.7	0.3w	7.05	+0.63	3.66	+0.22
Other β values					
0.5	0.5w	7.10	+0.68	3.68	+0.24
0.9	0.5w	7.09	+0.67	3.68	+0.24

Table 5 scans the region-mixing coefficient α and the region aggregation power β. The coefficient α controls the trade-off between local region advantages and the global advantage. When the region signal is too strong or introduced too early, updates can be dominated by local scores, weakening global quality constraints such as composition, identity preservation, and naturalness. The parameter β controls how sensitive region aggregation is to area: aggregation close to area weighting

Table 6: **Progressive ablation.** Starting from a Flow-GRPO baseline using the sample-level score from SFReward, we progressively add region advantages, global–region mixed advantages, α warmup, and region-balanced aggregation. The corresponding β/α settings are also shown.

Variant	Reg. adv.	Mixed adv.	α warm.	Reg. ratio	Bal. agg.	β	α	Δ GEdit	Δ ImgEdit
V1 (Flow-GRPO)	✗	✗	✗	✗	✗	–	–	0.57	0.22
V2 (+Reg. adv.)	✓	✗	✗	✓	✗	1	1	0.50	0.19
V3 (+Mix.)	✓	✓	✗	✓	✗	0	0.5	0.70	0.26
V4 (+Warmup)	✓	✓	✓	✓	✗	0	0.5w	0.75	0.25
V5 (Ours)	✓	✓	✓	✓	✓	0.7	0.5w	**0.87**	**0.30**

can still dilute foreground signals by large background regions, while equal region weighting can over-amplify high-variance small regions. These results suggest that region feedback should be introduced gradually and under global constraints.

Table 6 further shows that adding region advantages alone does not stably outperform Flow-GRPO, indicating that fine-grained rewards provide more localized supervision but can also amplify local noise without proper constraints. Global–region mixing provides an image-level quality anchor, α warmup prevents early training from relying too heavily on unstable local signals, and region-balanced aggregation reduces the dilution of foreground edits by background area. Overall, the benefit of SpatialFlow-GRPO comes from the cooperation of region rewards, global quality constraints, and the region-aligned policy objective, rather than from simply replacing sample-level rewards with region rewards.

5 Conclusion and discussion

We presented SpatialFlow-GRPO for spatial credit assignment in image editing. It treats semantic regions as fine-grained optimization units and uses SFReward and SFReward-14K to provide region-level feedback. On OmniGen2 and FLUX.2-klein-4B, SpatialFlow-GRPO consistently outperforms Flow-GRPO with the same reward model across three training seeds, with larger gains on MultiEditBench multi-region edits, showing that semantically grouped rewards better capture local editing quality. The method still relies on region annotations and reward-model reliability; future work can study more flexible region representations and larger-scale human evaluation.

Acknowledgments and Disclosure of Funding

Funding. This work received computational support from Kuaishou Technology.

Competing interests. The authors declare no competing financial interests beyond the affiliations listed in this paper.

References

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

[2] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning, 2024. URL https://arxiv.org/abs/2305.13301.

[3] Black Forest Labs. FLUX.2 [klein] 4B Base. Hugging Face model card, 2026. URL https://huggingface.co/black-forest-labs/FLUX.2-klein-base-4B.

[4] 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, 2025. URL https://arxiv.org/abs/2506.15742.

[5] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18392–18402, June 2023.

[6] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. DiffEdit: Diffusion-based semantic image editing with mask guidance, 2022. URL https://arxiv.org/abs/2210.11427.

[7] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining, 2025. URL https://arxiv.org/abs/2505.14683.

[8] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=FPnUhsQJ5B.

[9] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 79858–79885. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/fc65fab891d83433bd3c8d966edde311-Paper-Conference.pdf.

[10] 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, 2024. URL https://arxiv.org/abs/2309.17102.

[11] Google DeepMind. Gemini 3 Pro model card, 2025. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf.

[12] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen,

Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

[13] Yuxiang Guo, Jiang Liu, Ze Wang, Hao Chen, Ximeng Sun, Yang Zhao, Jialian Wu, Xiaodong Yu, Zicheng Liu, and Emad Barsoum. ImageDoctor: Diagnosing text-to-image generation via grounded image reasoning, 2025. URL https://arxiv.org/abs/2510.01010.

[14] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control, 2022. URL https://arxiv.org/abs/2208.01626.

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

[16] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

[17] Yushi Hu, Reyhane Askari-Hemmat, Melissa Hall, Emily Dinan, Luke Zettlemoyer, and Marjan Ghazvininejad. Multimodal RewardBench 2: Evaluating omni reward models for interleaved text and image, 2026. URL https://arxiv.org/abs/2512.16899.

[18] Zijing Hu, Fengda Zhang, Long Chen, Kun Kuang, Jiahui Li, Kaifeng Gao, Jun Xiao, Xin Wang, and Wenwu Zhu. Towards better alignment: Training diffusion models with reinforcement learning against sparse rewards. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23604–23614, June 2025.

[19] Mingzhen Huang, Jialing Cai, Shan Jia, Vishnu Suresh Lokhande, and Siwei Lyu. ParallelEdits: Efficient multi-object image editing, 2025. URL https://arxiv.org/abs/2406.00985.

[20] Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, and Ying Shan. SmartEdit: Exploring complex instruction-based image editing with multimodal large language models. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 8362–8371, June 2024.

[21] Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, Peng Wang, Yuyin Zhou, and Cihang Xie. HQ-Edit: A high-quality dataset for instruction-based image editing, 2024. URL https://arxiv.org/abs/2404.09990.

[22] 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 A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 36652–36663. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/73aacd8b3b05b4b503d58310b523553c-Paper-Conference.pdf.

[23] Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. VIEScore: Towards explainable metrics for conditional image synthesis evaluation, 2024. URL https://arxiv.org/abs/2312.14867.

[24] Fan Li, Chonghuinan Wang, Lina Lei, Yuping Qiu, Jiaqi Xu, Jiaxiu Jiang, Xinran Qin, Zhikai Chen, Fenglong Song, Zhixin Wang, Renjing Pei, and Wangmeng Zuo. HP-Edit: A human-preference post-training framework for image editing, 2026. URL https://arxiv.org/abs/2604.19406.

[25] Tiancheng Li, Jinxiu Liu, Huajun Chen, and Qi Liu. InstructRL4Pix: Training diffusion for image editing by reinforcement learning, 2024. URL https://arxiv.org/abs/2406.09973.

[26] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

[27] 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, 2025. URL https://arxiv.org/abs/2505.05470.

[28] 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, 2025. URL https://arxiv.org/abs/2504.17761.

[29] 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, 2026. URL https://arxiv.org/abs/2602.07458.

[30] Xin Luo, Jiahao Wang, Chenyuan Wu, Shitao Xiao, Xiyan Jiang, Defu Lian, Jiajun Zhang, Dong Liu, and Zheng Liu. EditScore: Unlocking online RL for image editing via high-fidelity reward modeling, 2026. URL https://arxiv.org/abs/2509.23909.

[31] Yiwei Ma, Jiayi Ji, Ke Ye, Weihuang Lin, Zhibin Wang, Yonghan Zheng, Qiang Zhou, Xiaoshuai Sun, and Rongrong Ji. I2EBench: A comprehensive benchmark for instruction-based image editing. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 41494–41516. Curran Associates, Inc., 2024. doi: 10.52202/079017-1313. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/48fecef47b19fe501d27d338b6d52582-Paper-Conference.pdf.

[32] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations, 2022. URL https://arxiv.org/abs/2108.01073.

[33] OpenAI. Introducing GPT-4.1 in the API, 2025. URL https://openai.com/index/gpt-4-1/.

[34] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4195–4205, October 2023.

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

[36] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

[37] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation, 2018. URL https://arxiv.org/abs/1506.02438.

[38] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

[39] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu Edit: Precise image editing via recognition and generation tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8871–8879, June 2024.

[40] Yichun Shi, Peng Wang, and Weilin Huang. SeedEdit: Align image re-generation to image editing, 2024. URL https://arxiv.org/abs/2411.06686.

[41] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. URL https://arxiv.org/abs/2011.13456.

[42] Zhenyu Tang, Chaoran Feng, Yufan Deng, Jie Wu, Xiaojie Li, Rui Wang, Yunpeng Chen, and Daquan Zhou. Enhancing spatial understanding in image generation via reward modeling, 2026. URL https://arxiv.org/abs/2602.24233.

[43] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8228–8238, June 2024.

[44] Cong Wei, Zheyang Xiong, Weiming Ren, Xeron Du, Ge Zhang, and Wenhu Chen. OmniEdit: Building image editing generalist models through specialist supervision. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Hlm0cga0sv.

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

[46] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, Ze Liu, Ziyi Xia, Chaofan Li, Haoge Deng, Jiahao Wang, Kun Luo, Bo Zhang, Defu Lian, Xinlong Wang, Zhongyuan Wang, Tiejun Huang, and Zheng Liu. OmniGen2: Towards instruction-aligned multimodal generation, 2026. URL https://arxiv.org/abs/2506.18871.

[47] Keming Wu, Sicong Jiang, Max Ku, Ping Nie, Minghao Liu, and Wenhu Chen. EditReward: A human-aligned reward model for instruction-guided image editing, 2026. URL https://arxiv.org/abs/2509.26346.

[48] 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, 2023. URL https://arxiv.org/abs/2306.09341.

[49] 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 A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 15903–15935. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/33646ef0ed554145eab65f6250fab0c9-Paper-Conference.pdf.

[50] Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, and Ping Luo. DanceGRPO: Unleashing GRPO on visual generation, 2025. URL https://arxiv.org/abs/2505.07818.

[51] Yankai Yang, Yancheng Long, Hongyang Wei, Wei Chen, Tianke Zhang, Kaiyu Jiang, Haonan Fan, Changyi Liu, Jiankang Chen, Kaiyu Tang, Bin Wen, Fan Yang, Tingting Gao, Han Li, and Shuo Yang. Joint reward modeling: Internalizing chain-of-thought for efficient visual reward models, 2026. URL https://arxiv.org/abs/2602.07533.

[52] 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, 2025. URL https://arxiv.org/abs/2505.20275.

[53] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

[54] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. MagicBrush: A manually annotated dataset for instruction-guided image editing. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 31428–31449. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/64008fa30cba9b4d1ab1bd3bd3d57d61-Paper-Datasets_and_Benchmarks.pdf.

[55] 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 A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 3058–3093. Curran Associates, Inc., 2024. doi: 10.52202/079017-0100. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/05a30a0fc9e6bacdd3abd4ca8508a9e6-Paper-Datasets_and_Benchmarks_Track.pdf.

A Additional details

A.1 Hyperparameters and training details

Table 7 lists the actual RL post-training configuration used for the main OmniGen2 experiments with SpatialFlow-GRPO. All RL post-training experiments use the 9-category editing subset of EditScore-RL-Data [30].

Table 7: **Main hyperparameters.**

Configuration	Value
Data and sampling	
RL training data	EditScore-RL-Data 9-category editing subset
Maximum text tokens	888
Output resolution cap	512×512
Unique instructions per round	48
Samples per instruction G	12
Inference timesteps	20
Maximum generation length	1024
Text/image guidance scale	4 / 2
CFG training interval	$[0.0, 0.6]$
Training timestep ratio	0.6
Optimization and finetuning	
Global batch size	576
Training batch size	36
Gradient accumulation steps	2
Total training steps	2000
Learning rate	4×10^{-4}
Learning-rate schedule	timm_constant_with_warmup
Optimizer	AdamW ($\beta_1 = 0.9, \beta_2 = 0.95$)
Weight decay / ϵ	0.01 / 10^{-8}
Maximum gradient norm	1
Mixed precision	bf16
Gradient checkpointing	Enabled
Finetuning strategy	LoRA [16] (r=32, α=64, dropout=0)
SpatialFlow-GRPO specific	
Update steps after sampling	2
Forward batch size	9
PPO clip range	$[10^{-4}, 5 \times 10^{-4}]$
Advantage clipping threshold	5
KL weight λ_{kl}	0.04
Sampling noise coefficient	0.7
Mixing coefficient $\alpha_{\max}$	0.5
Mixing coefficient $\alpha_{\min}$	0.0
Warmup start/end steps s_{start} / s_{end}	0 / 500
Region power β	0.7
Smoothing term τ	10^{-2}
Policy-loss reweighting	Enabled

A.2 Training algorithm

Algorithm 1 SpatialFlow-GRPO training

Require: Policy model θ, reference model θ_{ref}, region reward model $\mathcal{R}$, mixing parameter $\alpha(s)$, power parameter β

for each training round **do**
  Sample G edited images for each editing instruction p and record old-policy log probabilities
  Use $\mathcal{R}$ to evaluate structured region scores $\{R_{i,r}\}$
  Compute region advantages $\hat{A}_{i,r}$ by grouping over (p, l) (Equation 5)
  Compute the global advantage $\hat{A}_i^g$ and mix it into $\hat{A}_{i,r}^{\text{mix}}$ (Equation 6)
  Map $\hat{A}_{i,r}^{\text{mix}}$ to latent positions to form a fine-grained advantage map
  for each update step **do**
    Compute current-policy position-level log probabilities
    Compute the region-consistent ratio $s_{i,t,r}$ (Equation 9)
    Compute the power-weighted region policy loss $L_{\text{pg}}^{(t)}$ (Equation 11)
    Compute the KL regularization term $L_{\text{kl}}^{(t)}$
    Update θ using Equation (13)
  end for
end for

A.3 Region assignment and sparse semantic groups

Overlapping region boxes. SFReward predicts semantic bounding boxes, so different edited regions may overlap after being mapped to the latent resolution. For a latent position k, let $\mathcal{C}_i(k) = \{r \in \mathcal{R}_i^{\text{fg}} \mid k \in M_{i,r}\}$ denote the foreground regions that cover it. If $\mathcal{C}_i(k)$ is non-empty, the position-level mixed advantage is the average of $\hat{A}_{i,r}^{\text{mix}}$ over $r \in \mathcal{C}_i(k)$, and the corresponding region-ratio term is averaged over the same covering regions. If $\mathcal{C}_i(k)$ is empty, the position is assigned to the background region. This gives each latent position a unique and well-defined training signal while allowing overlapping positions to receive credit from all relevant semantic regions.

Sparse semantic labels. Equation (5) is used only when a semantic group contains at least two valid region rewards and has non-degenerate variance. If a label appears only once in the sampled group for an instruction, or if the within-group variance is too small, the group does not provide a reliable relative comparison. In this case, we do not compute an unstable region z-score; instead, the region advantage falls back to the corresponding sample-level global advantage $\hat{A}_i^g$. Thus the update reduces to global credit assignment when region-level comparison is underdetermined, and uses semantic-region relative advantages only when they are well defined.

A.4 SFReward and SFReward-14K details

A.4.1 SFReward-14K construction

The region-aware reward model used in our experiments is denoted as SFReward. To train this model, we build SFReward-14K from multi-source edit triplets. As shown in Figure 6, the pipeline collects a source image, an edited image, and an instruction, asks expert annotators to mark normalized bounding boxes and semantic labels for edited regions, uses Gemini-3-Pro [11] to produce per-region success/preserve scores, background preservation scores, overall scores, and reasoning, and then applies automatic structural validation followed by human audit. This pipeline explicitly binds reward scores to semantic regions, providing input signals for region advantage estimation and local policy updates in SpatialFlow-GRPO. SFReward-14K will be released to support reproduction of SFReward training and research on region rewards.

A.4.2 SFReward training setup

SFReward is initialized from Qwen3-VL-8B-Instruct [1] and trained on SFReward-14K with supervised finetuning. Table 8 gives the main training settings.

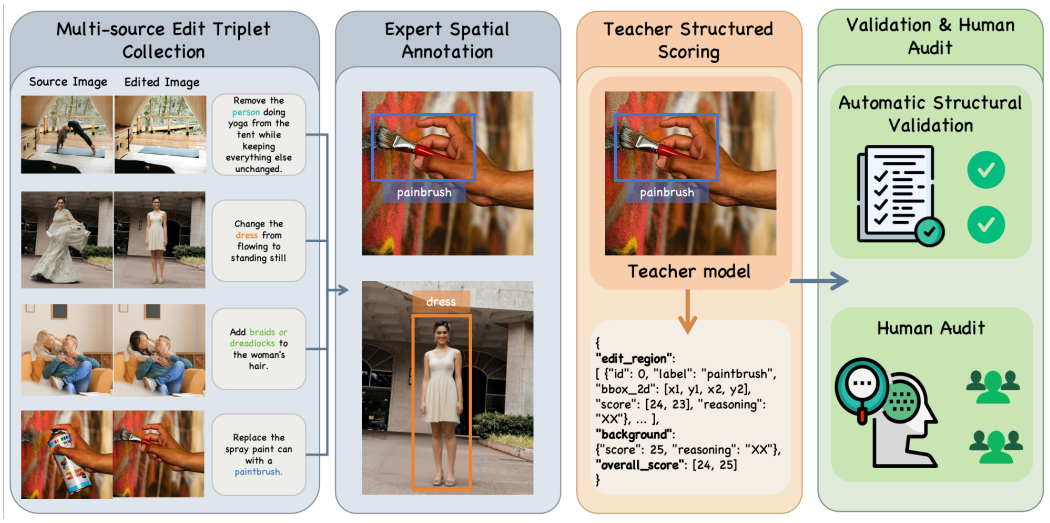

Figure 6: **SFReward-14K construction pipeline.** Multi-source edit triplets are annotated with expert region boxes and labels, scored by a multimodal teacher, and filtered through automatic validation and human audit to produce data for training a region-aware reward model.

Table 8: **SFReward training hyperparameters.**

Configuration	Value
Base model	Qwen3-VL-8B-Instruct
Training stage	SFT
Finetuning method	LoRA (r=32, α=64, dropout=0.05)
LoRA target modules	all
Training data	dense_grpo_coldstart_v1
Input template	qwen3_vl
Maximum sequence length	8192
Maximum image/video pixels	262144 / 16384
Per-device batch size	4
Gradient accumulation steps	4
Learning rate	1e-4
Training epochs	10
Learning-rate schedule	cosine, warmup ratio 0.1
Optimizer	AdamW
Training precision	bfloat16
Preprocessing/loading workers	192 / 48

A.4.3 Scoring prompt for the region-aware reward model

The following prompt is used to label SFReward-14K with the Gemini-3-Pro teacher model. It instructs the model to score each editing region in a structured and multi-dimensional way with reasoning.

```
You are a professional digital artist. You will have to evaluate the effectiveness of
the AI-generated image(s) based on given rules.
All the input images are AI-generated.
OUTPUT FORMAT:
{
  "edit_region": [
    {"id": 0, "label": "region label",
      "bbox_2d": [x1, y1, x2, y2],
      "score": [score_success, score_preserve],
      "reasoning": "brief reason"},
    ...
```

```
  ],
  "background":  {"score":  value, "reasoning":  "..."},
  "overall_score":  [overall_success, overall_preserve]
}

RULES:
Two images will be provided:  original and edited version.
The objective is to evaluate how successfully the editing instruction has been executed.
You will be provided with pre-identified editing regions (bounding boxes with labels).
Score each region separately.

SCORING (per region, 0-25):
1) score_success:  how well the edit follows the instruction (0=no change, 25=perfect).
2) score_preserve:  degree of preservation within the region (0=completely different,
25=minimal effective edit).

BACKGROUND (0-25):
Rate how well non-edited areas are preserved.  Penalize unexpected edits, layout changes,
artifacts outside editing regions.

OVERALL (0-25):
Overall success score and overall overediting score.
```

The prompt has three key design elements. First, **region-decoupled scoring** evaluates success and preserve scores independently for each edited region, while keeping whole-image scores as auxiliary information. Second, **spatial anchoring** binds scores to pre-annotated bounding boxes so that the scorer attends to the correct spatial location. Third, **reasoning** asks the scorer to provide a short explanation for each region, improving interpretability and consistency.

A.4.4 Example output from the region reward model

The following is a structured output example from the region-aware reward model for a multi-region editing instruction. The instruction is: "Change the man's shirt to red and add a tree in the background." SFReward first outputs semantic-consistency scores:

```
{
  "edit_region": [
    {
      "id": 0, "label": "person's shirt",
      "bbox_2d": [120, 80, 420, 360],
      "score": [22, 20],
      "reasoning": "Shirt color successfully changed to
        red; slight texture distortion at collar."
    },
    {
      "id": 1, "label": "tree",
      "bbox_2d": [520, 100, 860, 700],
      "score": [18, 15],
      "reasoning": "Tree added but appears slightly
        unnatural; some blending artifacts with sky."
    }
  ],
  "background": {
    "score": 21,
    "reasoning": "Background well preserved; minor
      color shift near added tree boundary."
  },
  "overall_score": [20, 18]
}
```

At the same time, SFReward outputs perceptual-quality scores:

```
{
  "reasoning": "The edited image is mostly natural,
```

```
    with mild artifacts around the inserted tree.",
  "score": [22, 20]
}
```

The above output is converted into scalar region rewards by Equation (3). In this example, the `shirt` region receives a high reward because the edit succeeds and is well preserved. The `tree` region receives a lower reward because artifacts appear. The background region receives a moderate reward. Differentiated region rewards provide the basis for local credit assignment in SpatialFlow-GRPO.

A.4.5 Training data statistics

Table 9 reports statistics of SFReward-14K. The data is filtered through multiple rounds of quality control, resulting in a final retention rate of 98.75%.

Table 9: **SFReward-14K statistics.**

Statistic	Value
Total samples	14,276
Average edited regions per sample	2.32
Maximum edited regions per sample	7
Single-region samples	24.3%
Multi-region samples	75.7%
Edit type distribution	
Attribute change (color/material/texture)	38.2%
Object operation (add/replace/remove)	33.5%
Background modification	18.7%
Style/global transformation	9.6%
Quality control	
Initially collected samples	16,465
Structured-output pass rate	99.68%
Final retention rate	98.75%

A.5 MultiEditBench details

A.5.1 Data construction and statistics

MultiEditBench is designed to diagnose model performance in multi-region parallel editing scenarios. Source images come from several high-quality public sources and cover visual types such as people, indoor and outdoor scenes, object close-ups, and artistic compositions. The source distribution is shown in Table 10. For each source image, we first use a VLM to identify editable elements. An LLM then generates compound instructions containing 2–5 edits. Human review removes ambiguous, contradictory, or overly simple samples.

Table 10: **Source images in MultiEditBench.**

Source	Ratio	Description
LAION-Aesthetics	42.4%	Natural images with high aesthetic scores
CC8M	42.0%	Diverse web images
Pexels	7.8%	Professional photography images
Others	7.8%	Other high-quality sources

Editing operations include attribute modification, object replacement, element addition, enhancement, removal, and position adjustment. Since a compound instruction usually contains multiple operation types, the frequencies in Table 11 are not mutually exclusive.

Table 11: **Editing operation types in MultiEditBench.**

Operation type	Frequency	Example
Change (attribute modification)	69.8%	Change color, material, or expression
Replace (object replacement)	58.8%	Replace an object with another object
Add (element addition)	49.4%	Add a new object or decoration
Enhance (enhancement)	27.0%	Enhance lighting, makeup, or details
Remove (removal)	15.2%	Remove a specified element
Adjust (adjustment)	13.2%	Adjust pose, angle, or position

Table 12: **Difficulty distribution of MultiEditBench.** Samples are grouped by the number of simultaneous editing operations.

Type	Meaning	Count	Ratio	Avg. instruction length
2-edit	2 simultaneous edits	140	28.0%	131 chars
3-edit	3 simultaneous edits	165	33.0%	169 chars
4-edit	4 simultaneous edits	150	30.0%	198 chars
5-edit	5 simultaneous edits	45	9.0%	256 chars
Total	–	500	100%	–

A.5.2 MultiEditBench evaluation prompts

MultiEditBench follows the VIEScore [23] framework and uses GPT-4.1 [33] as the evaluation backbone. Each sample is evaluated separately for semantic consistency (SC) and perceptual quality (PQ), and the final metric is MEB Score $= \sqrt{\text{SC} \times \text{PQ}}$. Scores are averaged within each task type. The final Avg is the equal-weighted average over the 2-edit, 3-edit, 4-edit, and 5-edit difficulty subsets.

Semantics Score (SC). SC takes the source image, edited image, and editing instruction as input. It outputs editing success and over-editing control scores, and their mean is used as SC.

```
You are a professional digital artist. You will evaluate the effectiveness of the
AI-generated image(s).

RULES:
Two images will be provided: the first being the original AI-generated image and the
second being an edited version.
Evaluate how successfully the editing instruction has been executed in the second image.

SCORING (0-10):
score1: editing success, where 0 means the edit does not follow the instruction and 10
means perfect instruction following.
score2: degree of overediting, where 0 means the edited image is completely different
from the original and 10 means a minimal yet effective edit.
Output: {"score": [score1, score2], "reasoning": "..."}

Editing instruction: {instruction}
```

Quality Score (PQ). PQ takes only the edited image as input. It outputs naturalness and artifact-control scores, and their mean is used as PQ.

```
The image is an AI-generated image. Evaluate the generation quality.

SCORING (0-10):
naturalness: 0 means the image does not look natural; 10 means it looks natural.
artifacts: 0 means severe distortion, watermark, blur, unusual body parts, or
disharmonized subjects; 10 means no artifacts.
Output: {"score": [naturalness, artifacts], "reasoning": "..."}
```

A.6 Notation

Table 13 summarizes the main notation used in this paper.

Table 13: **Main notation.**

Symbol	Meaning
G	Number of samples per editing instruction
T	Total number of diffusion timesteps
c	Editing instruction (condition)
$R_{i,r}$	Region reward of region r in sample i
$\hat{A}_{i,r}$	Region advantage of region r in sample i
$\hat{A}^{\text{mix}}_{i,r}$	Global–region mixed advantage
$\alpha(s)$	Region mixing coefficient as a function of training step s
β	Region power-weighting parameter
τ	Smoothing term for power weighting
$M_{i,r}$	Latent-position set corresponding to region r
$s_{i,t,r}$	Region-consistent ratio
$\mathcal{S}_{p,l}$	Semantic group (instruction $= p$, label $= l$)
ϵ_+, ϵ_-	PPO clipping bounds
λ_{kl}	KL regularization weight
ω_t	Timestep reweighting coefficient

A.7 Statistical evaluation and compute resources

For the main RL comparisons, we train Flow-GRPO and SpatialFlow-GRPO on both base models with three random seeds and report the mean, sample standard deviation, and mean paired gain in Table 3. All other RL results are from single training runs.

The main RL post-training experiments were run on 16 NVIDIA H800 GPUs for about 40 hours, corresponding to roughly 640 GPU-hours for a main run. The same hardware class was used for Flow-GRPO baselines and SpatialFlow-GRPO variants. Compute for ablations scales approximately linearly with the number of training variants because each variant uses the same sampling and update schedule.

A.8 Responsible research, assets, and LLM usage

This work improves image editing models and may have both positive and negative societal impacts. Positive uses include controllable content creation, image restoration, and editing assistance; negative uses include misleading edits, identity manipulation, and disinformation. Released data and models will include usage restrictions, safety notes, and filtering for unsafe or privacy-sensitive content.

All existing models, datasets, and benchmarks are cited in the main text or bibliography. We follow their licenses and terms of use, and will provide the code in the supplementary material.

Human annotation is used to mark editing regions, assign semantic labels, and review benchmark samples. Annotators receive written instructions and examples, are compensated according to local requirements, and the task does not collect private information or involve interventions on human subjects.

LLMs and VLMs are used in the data and evaluation pipeline: Gemini-3-Pro produces structured region scores for SFReward-14K, VLM/LLM tools identify editable elements and generate candidate MultiEditBench instructions before human review, and GPT-4.1 serves as the VIEScore-style evaluation backbone. These uses are described in Appendix A.4.1 and Appendix A.5.2.

A.9 Additional OmniGen2 qualitative comparisons

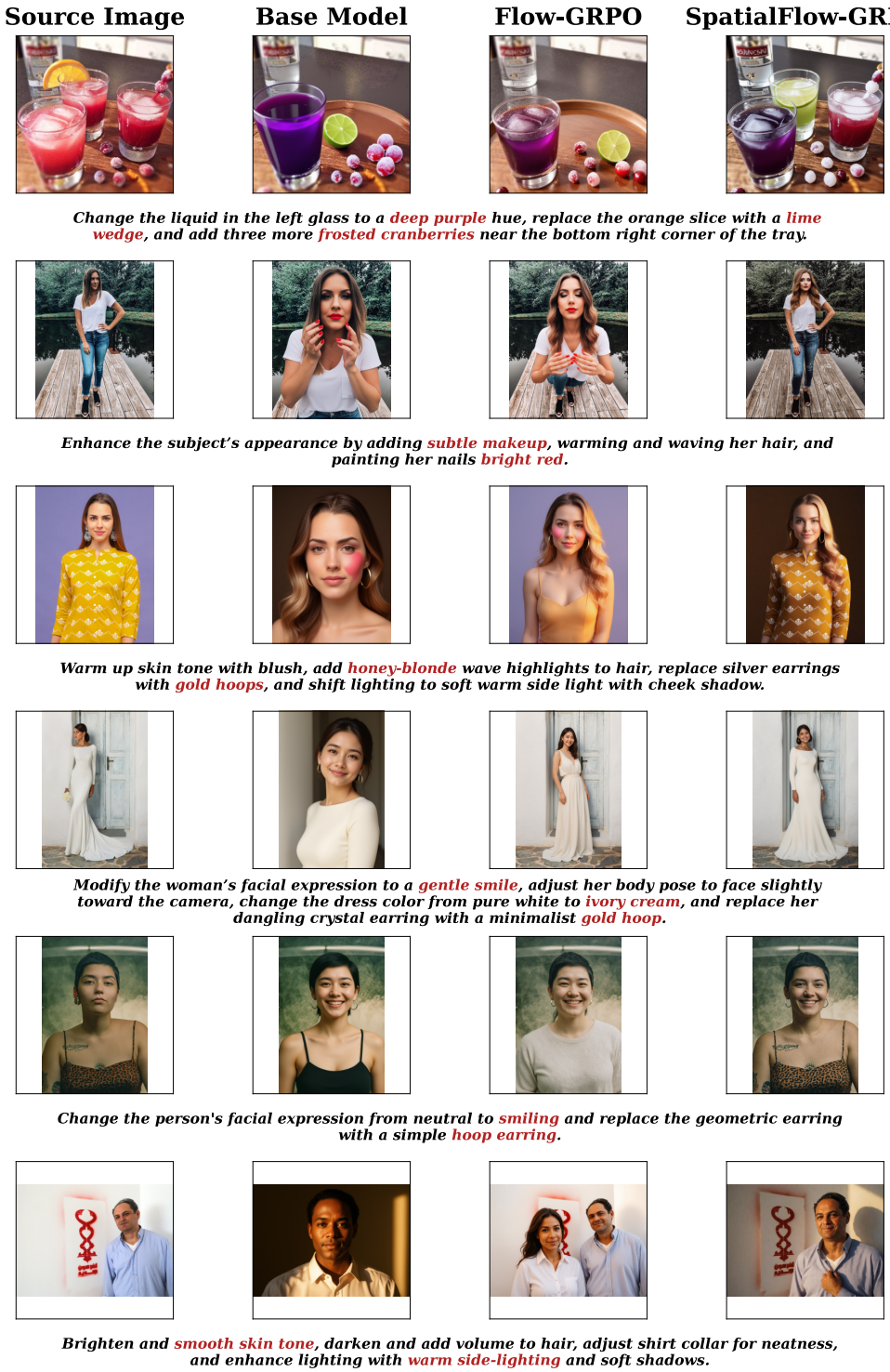

Figure 7: **Additional OmniGen2 qualitative comparisons** (1/3). Each row compares the base model, Flow-GRPO, and SpatialFlow-GRPO under multi-target editing instructions.

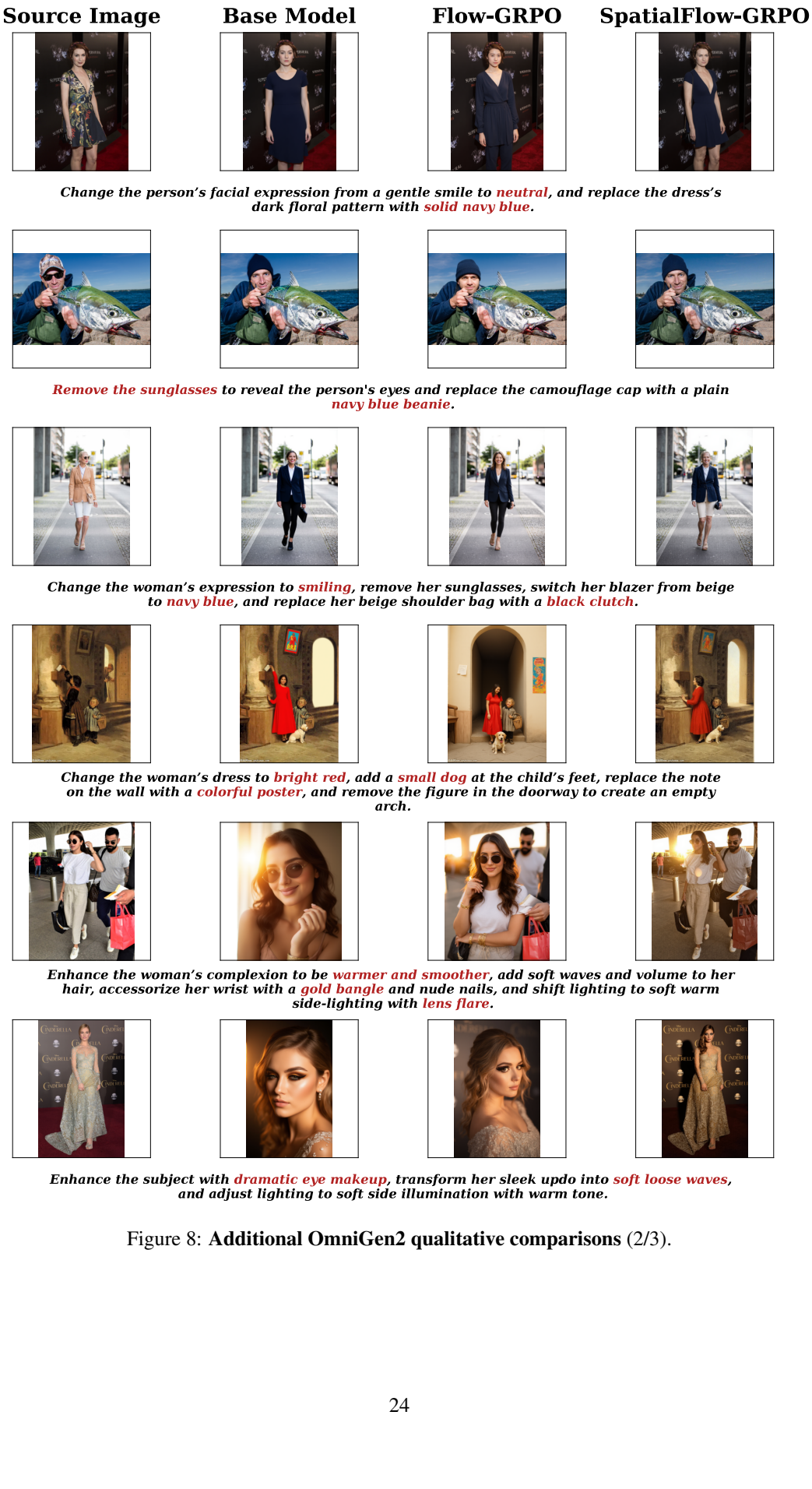

Figure 8: **Additional OmniGen2 qualitative comparisons** (2/3).

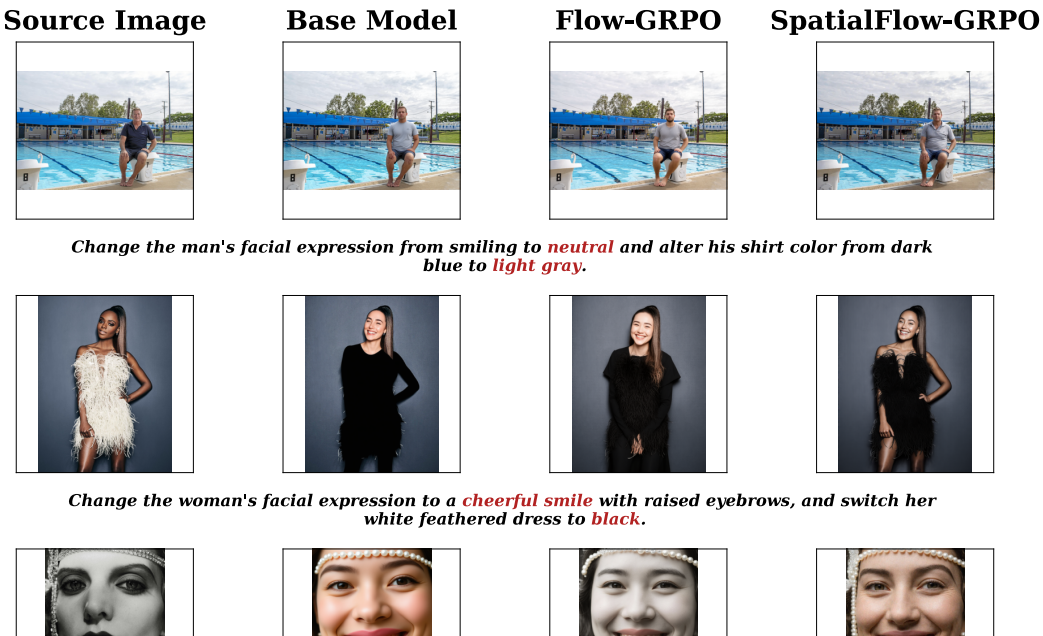

Figure 9: **Additional OmniGen2 qualitative comparisons** (3/3).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and Introduction state the scope, contributions, assumptions, and evaluated settings. The claims are supported by the method description and experiments in Sections 3–4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: The paper discusses limitations in Section 5, including dependence on region annotations, reward-model quality, and the need for larger-scale human evaluation.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A].

Justification: The paper does not present formal theoretical results or theorems; it proposes and empirically evaluates an RL post-training algorithm.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes].

Justification: The method, objective functions, training algorithm, datasets, evaluation protocols, and hyperparameters are described in Sections 3–4 and Appendix A.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: Code will be provided in the supplementary material, as described in Appendix A.8.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Experimental settings are specified in Section 4.1; training hyperparameters, data construction, evaluation prompts, and benchmark statistics are provided in Appendix A.1–A.5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: Table 3 reports the mean $\pm$ 1-sigma sample standard deviation over three training seeds for the main Flow-GRPO vs. SpatialFlow-GRPO comparisons on both base models, together with mean paired gains. The variability captured is that of independent training runs; Appendix A.7 describes this setting.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Appendix A.7 reports the compute resources used for the main RL post-training experiments, including the use of 16 NVIDIA H800 GPUs for about 40 hours.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The research follows the NeurIPS Code of Ethics. Responsible use, asset handling, human annotation, and release safeguards are discussed in Appendix A.8.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

Justification: Appendix A.8 discusses positive uses such as controllable editing and possible negative uses such as misleading edits, identity manipulation, and disinformation.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [Yes].

Justification: Appendix A.8 describes responsible release measures, including usage restrictions, safety notes, and filtering procedures for unsafe or privacy-sensitive content.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes].

Justification: Existing models, datasets, and benchmarks are cited throughout the paper. Appendix A.8 states that the corresponding licenses and terms of use are followed.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: New assets, including SFReward-14K, SFReward, and MultiEditBench, are documented in Appendix A.4–A.5; code availability is described in Appendix A.8.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes].

    Justification: Human annotation is used for region labels and benchmark review. Appendix A.8 describes written instructions, examples, compensation, and the absence of private annotator-data collection.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

Justification: The work uses human annotation for region labeling and benchmark review, but does not conduct human-subject experiments, collect private participant information, or intervene on participants. Annotation instructions and compensation are discussed in Appendix A.8.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes].

Justification: Appendix A.8 declares LLM/VLM usage. Gemini-3-Pro is used for region-score generation, VLM/LLM tools are used for MultiEditBench construction, and GPT-4.1 is used for VIEScore-style evaluation.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.