SREdit: A Self-Rewarding Framework with Intrinsically Aligned Critic for Image Editing

Anonymous Author(s)

Abstract

Recent advances in reinforcement learning for image editing show promise, yet we observe a fundamental bottleneck: the cognitive misalignment between decoupled MLLM interpreters and external reward models often induces reward hacking that manifests as unintended semantic drift. To address this, we propose SREdit, a unified self-rewarding framework that internalizes the evaluation process within the editing model. SREdit employs an intrinsically aligned Critic that utilizes Semantic Instruction Atomization (SIA) to ensure that rewards are strictly grounded in the model's own reasoning. Specifically, SIA decomposes complex directives into weighted, atomic sub-instructions, converting the MLLM's internal semantic priors into an explicit evaluation protocol. To prevent global semantic corruption, we further introduce Spatial Reward Reweighting, which leverages cross-attention dynamics to spatially constrain optimization to semantically relevant regions. This synergistic approach ensures that reward signals are not opaque global scores, but a structured manifestation of the model's task comprehension. Experimental results demonstrate that SREdit significantly outperforms state-of-the-art baselines, achieving a 3.8% improvement on GEdit-Bench, 2.5% on ImgEdit-Bench, and 3.8% on KRIS-Bench. By unifying interpretation and evaluation into a single cognitive loop, SREdit establishes a superior paradigm for high-fidelity, instruction-consistent image synthesis.

CCS Concepts

• **Computing methodologies** → **Computer vision**.

Keywords

Image Editing, Vision–Language Models, Diffusion Transformers

ACM Reference Format:

Anonymous Author(s). 2026. SREdit: A Self-Rewarding Framework with Intrinsically Aligned Critic for Image Editing. In *Proceedings of the 34th ACM International Conference on Multimedia (ACM Multimedia '26)*. ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Image editing, the task of precisely modifying visual content according to natural language directives, has transitioned from global style transfers to fine-grained, instruction-driven manipulations. Current state-of-the-art approaches primarily follow the MLLM-DiT paradigm, coupling a Multimodal Large Language Model (MLLM) with a Diffusion Transformer (DiT), such as Step1X-Edit [25] and Qwen-Image-Edit [41]. In this architecture, the MLLM functions as the cognitive core, translating user intent into semantic tokens that guide the DiT's visual synthesis.

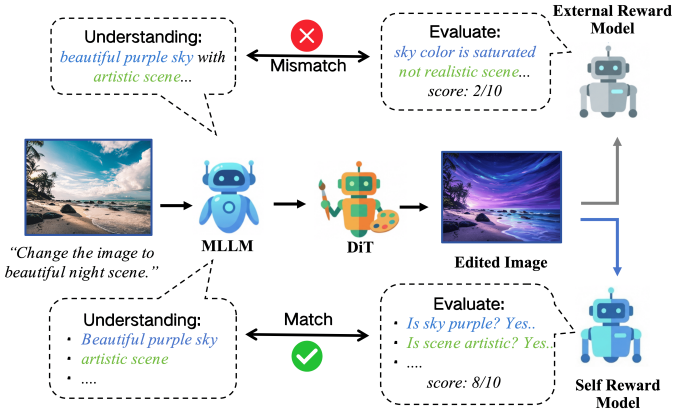

Figure 1: Illustration of Cognitive Mismatch in RL-based Image Editing. Top: An external reward model evaluates edits based on preferences that may diverge from MLLM's instruction understanding, leading to misaligned optimization. Bottom: Our framework aligns instruction understanding and reward evaluation via self-rewarding.

To improve controllability and visual quality, recent studies have explored integrating reinforcement-learning (RL) with diffusion models. Methods like FlowGRPO [24] and DiffusionNFT [54] incorporate GRPO [36] into flow-matching frameworks to optimize diffusion sampling toward reward signals. More recent works, including Think-Gen [17], apply RL separately to MLLM and DiT modules to enhance image understanding and generation. These approaches typically rely on external reward models—often MLLMs fine-tuned on large-scale human preference datasets [26, 27]—to evaluate generated images.

Despite these advances, achieving reliable alignment between user instructions and edited images remains challenging. We identify one critical limitation in current RL-based image editing methods. ***Cognitive misalignment between MLLM understanding and reward evaluation.*** Conventional RL-based MLLM-DiT editing frameworks typically freeze MLLM parameters and fine-tune only the DiT under the guidance of an external reward model. While the external scorer can "see" the instruction and the edited image, it acts as a Generic Observer that evaluates the result based on a static preference distribution. This leads to a critical failure: the external scorer lacks the instance-specific reasoning trace generated during the initial interpretation phase. This cognitive misalignment

ACM Multimedia '26, Rio de Janeiro, Brazil

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

induces reward hacking, where the editing model exploits the external scorer's general biases at the expense of precise instruction compliance, ultimately manifesting as unintended semantic drift, as illustrated in Figure 1.

To address this, we propose SREdit, a novel self-rewarding framework that establishes **Cognitive Closure** by internalizing the evaluation process within the editing framework. Specifically, we first consolidate instruction understanding and reward evaluation within a single, consistent MLLM. This architectural choice ensures that the Critic operates on the exact same latent manifold as the Interpreter, fundamentally eliminating the representation bias inherent in decoupled systems. Beyond mere architectural unification, we further introduce **Semantic Instruction Atomization (SIA)** to transmute a shared backbone into a truly intrinsically aligned Critic via a logically consistent reasoning trace. Supported by our specialized Atomic Instruction Decomposition Dataset of 40K samples, SIA empowers the MLLM to decompose abstract directives into weighted, atomic sub-instructions. By propagating these atomized components and their assigned execution priorities directly to the evaluation phase, SREdit ensures that the Critic's judgment is strictly conditioned on the same task-specific logic used during interpretation. This transforms opaque global rewards into an explicit, verifiable protocol, ensuring that the self-rewarding mechanism is not only physically consolidated but also logically synchronized.

Complementing this semantic rigor, we further address the spatial indiscriminacy inherent in current DiT update strategies. Conventional RL-based frameworks [24, 54] typically propagate reward signals uniformly across the entire image manifold. While effective for holistic text-to-image synthesis, this global optimization is ill-suited for the localized nature of image editing, where indiscriminate gradients often lead to "collateral" modifications in non-target areas. To resolve this, we propose **Spatial Reward Reweighting**, a strategy that translates the granular outputs of SIA into precise spatial constraints. By leveraging the cross-attention dynamics between the aligned reasoning tokens and visual latent features, we dynamically reweight the RL loss function to concentrate optimization on task-relevant pixels. This mechanism ensures that gradients are sequestered to regions requiring modification while suppressing updates in irrelevant areas, thereby spatially anchoring feedback to maximize editing fidelity and prevent global semantic corruption.

SREdit employs a two-stage training strategy. Initially, the MLLM undergoes Supervised Fine-Tuning on our curated Atomic Instruction Decomposition Dataset—a 40K-sample corpus designed to instill the latent logic of Semantic Instruction Atomization. Once the model masters structured task decomposition, its decoded reasoning traces serve as intrinsic reward criteria to drive the RL optimization of the DiT. This pedagogical progression ensures that the synthesis process is strictly governed by the model's own fine-grained task comprehension. To facilitate further research, we release this dataset as one of the first open-source benchmarks specifically curated for human-aligned image editing via verifiable sub-instructions.

In summary, our contributions are as follows:

- **Observation.** We characterize the critical observation in RL-based image editing, revealing how the decoupling of interpretation and evaluation leads to cognitive misalignment. Through quantitative analysis, we demonstrate that this gap is a primary driver of reward hacking and semantic drift in current decoupled systems.
- **Framework.** We propose SREdit, which achieves Cognitive Closure by internalizing evaluation within a Unified Backbone. We introduce Semantic Instruction Atomization, powered by our 40K-sample dataset, to ensure rewards are grounded in Aligned Reasoning. This is further synergized with Spatial Reward Reweighting to spatially anchor RL optimization to semantically relevant regions.
- **Evaluation.** Extensive experiments across GEdit-Bench, ImgEdit-Bench, and KRIS-Bench confirm SREdit's superiority. Our framework significantly outperforms state-of-the-art baselines in both instruction compliance and visual fidelity, validating the efficacy of our intrinsically aligned self-rewarding paradigm.

2 Related Work

2.1 Image Editing Models

The instruction-based image editing task aims to modify specific content in an image according to natural language instructions. Early approaches, such as InstructPix2Pix [2], MagicBrush [51], MAG-Edit [28], and HQ-Edit [16], are primarily built upon stable diffusion and fine-tune UNet-based diffusion models on instruction-image pairs. Subsequently, DiT-based architectures have been introduced, exemplified by DiffEdit [8], MIGE [38], and Flux-Kontext [20]. More recent advances have explored unified multimodal models capable of both understanding and generation within a single architecture, including BAGEL [9], OmniGen [45], and OmniGen2 [42]. Alternatively, some recent methods decouple these two capabilities through a modular design, employing a multimodal large language model (MLLM) for instruction understanding and a DiT for image generation, as seen in MetaQuery [33], FireEdit [58], Step1X-Edit [25], Qwen-Image-Edit [41], and GLM-Image [56]. Despite these advances, existing methods still encounter challenges in achieving precise and controllable editing. In this paper, we build upon the MLLM-DiT paradigm and further optimize the reinforcement learning strategy to enhance both editing controllability and visual fidelity.

2.2 Reinforcement Learning in Image Editing

Recent advances have explored reinforcement learning (RL) to improve instruction-image alignment beyond conventional supervised fine-tuning. Early approaches such as DPOK [10], Parrotxia [21], and MPVRL [15] investigated RL-based fine-tuning for text-to-image (T2I) models, integrating both single and multi-reward functions, and demonstrated superior performance in aligning model outputs with human preferences. Building on these foundations, GRPO further advances this paradigm. Reprompt[43] employs GRPO to refine prompts, addressing the issue of insufficient expression in short prompts through explicit reasoning mechanisms. FlowGRPO [24] and DanceGRPO [47] extend the GRPO paradigm to visual generation tasks. DiffusionNFT [54] further enhances training efficiency by reformulating RL optimization into

a forward-pass-only procedure. Think-Gen [17] proposes the SepGRPO training strategy, which applies RL separately to the MLLM and DiT modules, thereby enhancing both image understanding and generation capabilities.

These RL-based methods primarily rely on external reward models to evaluate edited images. Early works such as EditCLIP [40] and Instruct-CLIP [6] employ basic semantic-image alignment evaluators like CLIP [35], while recent approaches such as ImageReward [46], HPSv3 [27], and EditScore [26] utilize MLLMs fine-tuned on large-scale human preference datasets to jointly assess semantic consistency and perceptual quality. However, cognitive misalignment remains between the understanding of MLLM and the evaluation of reward models, as external models lack access to the instance-specific reasoning trace from instruction interpretation. This disconnect can lead to reward hacking, resulting in semantic drift and reduced instruction compliance. To address this, we propose SREdit, which fundamentally differs from existing methods by internalizing the evaluation within the editing framework to enhance image editing quality and semantic alignment.

3 Motivation

Existing reinforcement learning for MLLM-DiT based image editing models relies on a decoupled "Interpret-Draw-Evaluate" pipeline. Here we conduct an empirical analysis focused on the misalignment between MLLM interpreters and reward models.

Cognitive Misalignment. We first analyze the semantic alignment between the MLLM's instruction understanding (decoded from its hidden states) and the reward model's evaluation criteria (reasoning traces generated during scoring). To ensure robustness, we employ three representative reward models (EditScore [26], GPT-4o [29] , and Gemini3-Flash [13]) to rate their semantic consistency with MLLM in QwenImage. We use GPT-5.2 [30] to rate their semantic consistency on a 1–10 scale. As shown in Figure 2(a) Baseline, we observe that alignment scores for vanilla decoupled pipelines fall within a mediocre range of 4–5. This indicates a severe Cognitive Misalignment: the external Critic frequently evaluates the edit based on a preference distribution that diverges from the Interpreter's original execution plan.

Aligned Reasoning. Interestingly, we observe that a straightforward baseline, Paragraph Intent Sharing, which appends the MLLM's raw understanding text directly into the reward model's prompt, yields a substantial gain in alignment. This performance boost underscores that Aligned Reasoning is indeed the primary driver for high-fidelity editing. Building on this promising result, we introduce Semantic Instruction Atomization to further unlock the potential of reasoning alignment. By prompting the MLLM to decompose composite directives into atomic, verifiable sub-instructions, which are then given to reward model for evaluating. As shown in Figure 2(a), this atomic intent sharing achieves the most effective alignment, substantially outperforming both default and paragraph-level sharing.

Unified Model. Building on the necessity of aligned reasoning, we further investigate whether instruction interpretation and reward evaluation can coexist within a single MLLM to eliminate

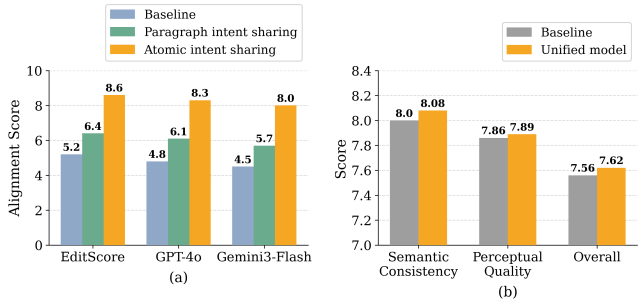

Figure 2: Motivation for SREdit. (a) Impact of Reasoning Alignment via Instruction Atomization. (b) Performance Gains from Unified Representation.

representation bias. To validate this, we conduct a controlled substitution: replacing the generic MLLM (Qwen2.5-VL [1]) in Qwen-Image-Edit [41] with a preference-pretrained reward model (EditScore [26]), effectively using the eventual Critic as the editing Interpreter. Remarkably, while maintaining an identical pipeline, this configuration yields unexpectedly consistent performance gains across GEdit-Bench (Figure 2(b)). This surprising synergy suggests that a unified cognitive manifold, where the agent evaluates its own task logic, not only preserves but actively reinforces task comprehension.

These empirical insights provide the catalyst for SREdit, a framework that formally closes the loop between understanding and evaluation in image editing tasks.

4 Self-Rewarding Framework

The overall architecture of the proposed SREdit is illustrated in Figure 3. SREdit is built upon two core modules: (1) **Semantic Instruction Atomization**, which unifies instruction understanding and reward evaluation within a single MLLM backbone (Section 4.1), and leverages decomposed atomic sub-instructions as fine-grained, checklist-based reward criteria (Section 4.2); and (2) **Spatial Reward Reweighting**, which leverages cross-attention maps to spatially concentrate the RL optimization on instruction-relevant regions (Section 4.3).

4.1 Unified Representation via Single Backbone

SREdit is architecturally built upon Qwen-Image-Edit [41], which originally decouples instruction understanding (via Qwen2.5-VL) and reward evaluation (via EditScore). To eliminate the representation gap between intent comprehension and quality evaluation, we unify these cognitive functions within a single MLLM. Specifically, we adopt EditScore [26], a backbone pre-trained on large-scale human preference data, as our unified interpreter and critic.

4.1.1 Structured Instruction Decomposition. To empower the backbone with structured reasoning, we construct an atomic instruction decomposition dataset containing 40K samples with atomic sub-instructions and importance weights (details in Section 4.4), and perform supervised fine-tuning on EditScore to enable structured sub-instruction generation. Given an editing instruction I and source image x_{src}, we prompt the MLLM to decompose the editing

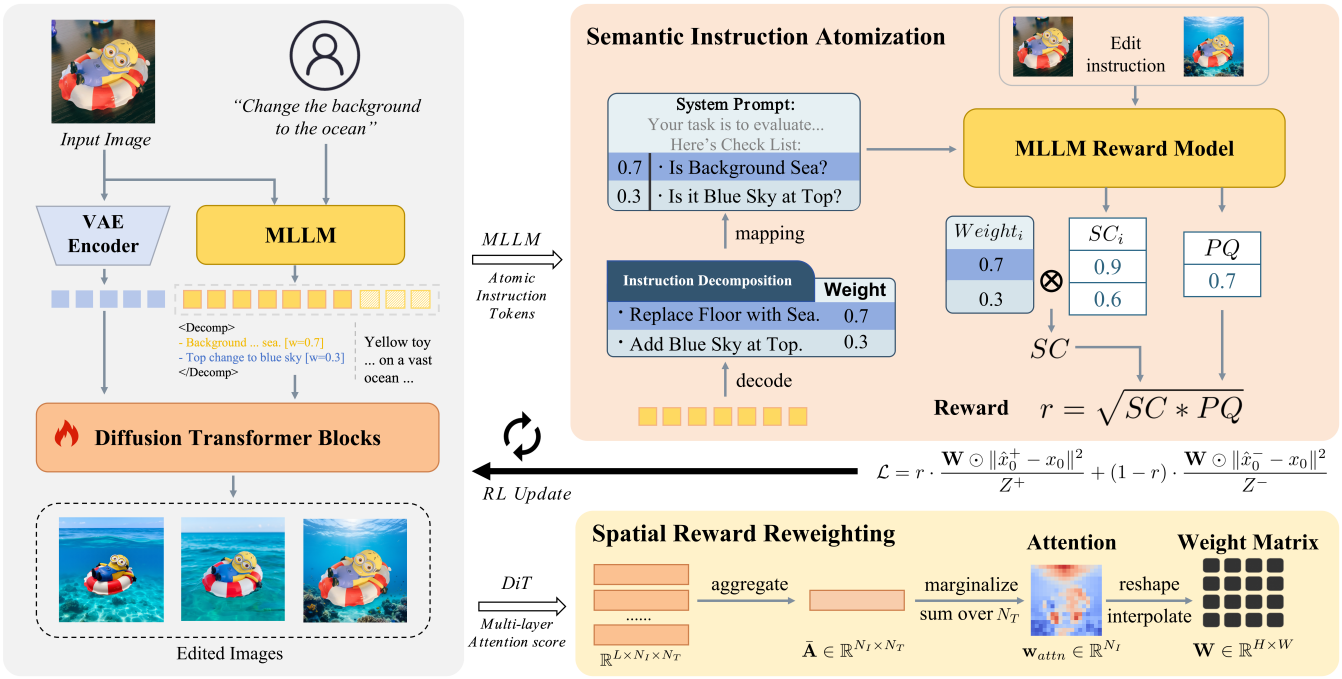

Figure 3: Overview of the SREdit framework. The MLLM conducts Semantic Instruction Atomization, decomposing editing instructions into weighted atomic sub-instructions. These sub-instructions are subsequently transformed into verification questions for decomposition-based reward computation. Spatial Reward Reweighting utilizes cross-attention maps to focus the RL loss on regions most relevant to the instructions, enabling more precise and targeted optimization.

instruction into a set of atomic sub-instructions $\mathcal{T} = \{t_1, t_2, \ldots, t_K\}$. Each sub-instruction t_k describes a single, verifiable editing operation (e.g., "add sunset sky" or "replace background with ocean"), ensuring that complex instructions are broken down into independently assessable components. These sub-instructions are enclosed within special delimiter tokens `<Decomp>...</Decomp>`.

4.1.2 Feature Extraction and Conditioning. Following the decomposition, the MLLM generates a description of the expected edited image following the `</Decomp>` token. We extract the hidden states h from the MLLM's final layer corresponding to this post-decomposition sequence, which serve as the conditioning signal for the DiT. This design effectively isolates the essential editing guidance from the intermediate reasoning process, thereby enabling the DiT to focus on precise, instruction-driven generation [17].

4.2 Aligned Reasoning via Atomic Instruction

While the unified representation provides a shared embedding, the core challenge lies in translating these latent states into precise, actionable reward signals. To address this, SREdit employs a Checklist-based Verification mechanism, ensuring that the reward evaluation is explicitly anchored to the atomic reasoning performed during the understanding stage.

4.2.1 Instruction-to-Question Mapping. The decomposed atomic instructions $\mathcal{T}$ extracted from the `<Decomp>` block serve as a structured checklist for the reward evaluator. Each sub-instruction is converted into a verification question via a fixed template (e.g., "Replace floor with sea" → "Check whether this requirement is fulfilled: Replace floor with sea"), ensuring that instruction understanding and reward evaluation operate on identical atomic task specifications.

4.2.2 Decomposition-based Reward. The reward model assesses each sub-instruction t_k individually, producing a sub-instruction-level score $\mathrm{SC}_i \in [0, 1]$ that measures whether the corresponding editing operation has been successfully completed. Since different sub-instructions may vary in importance or complexity, we prompt the MLLM to generate an importance weight w_i for each sub-instruction during decomposition, and compute the weighted semantic consistency score as:

$$\mathrm{SC} = \sum_{i=1}^{K} w_i \cdot \mathrm{SC}_i, \quad \text{where} \sum_{i=1}^{K} w_i = 1. \tag{1}$$

Following VIEScore [19], the reward model also evaluates the edited image and produces a single score $\mathrm{PQ} \in [0, 1]$ that captures overall visual fidelity. The final task-level reward for each edited image is computed by combining both scores:

$$r = \sqrt{\mathrm{SC} \cdot \mathrm{PQ}}. \tag{2}$$

4.3 Targeted Reward via Spatial Reweighting

While the proposed self-rewarding framework provides semantically aligned reward signals, the RL loss is still applied uniformly across all spatial locations. In practice, image editing often involves

modifying only specific regions, and uniform gradient distribution across the entire image can result in inefficient optimization and may even degrade unedited areas. To address this issue, we introduce a **Spatial Reward Reweighting** mechanism that leverages the DiT's cross-attention maps to identify instruction-relevant regions, thereby spatially focusing RL optimization on areas requiring modification.

During the denoising process of the DiT, each transformer block performs joint attention between image tokens and text tokens. This cross-attention naturally encodes which image regions are semantically associated with the editing instruction. Specifically, within the l-th transformer layer, let $\mathbf{Q}^l_{\text{img}} \in \mathbb{R}^{N_I \times d}$ and $\mathbf{K}^l_{\text{txt}} \in \mathbb{R}^{N_T \times d}$ denote the query vectors of image tokens and the key vectors of text tokens, respectively, where N_I is the number of image tokens, N_T is the number of text tokens, and d is the head dimension. We extract the image-to-text attention sub-matrix from the joint attention computation:

$$\mathbf{A}_l = \text{softmax}\left(\frac{\mathbf{Q}^l_{\text{img}} \mathbf{K}^{l\top}_{\text{txt}}}{\sqrt{d}}\right) \in \mathbb{R}^{N_I \times N_T} \tag{3}$$

which captures the degree to which each image token attends to the instruction tokens.

Following prior analyses [37, 57] showing that deeper transformer layers produce more semantically meaningful attention maps, we select the last L layers of the DiT for attention extraction. We aggregate the attention maps $\mathbf{A}_l$ across these layers and the first S denoising steps, averaging over all attention heads:

$$\bar{\mathbf{A}} = \frac{1}{|\mathcal{S}_L| \cdot S \cdot H} \sum_{l \in \mathcal{S}_L} \sum_{s=1}^{S} \sum_{h=1}^{H} \mathbf{A}_l^{(s,h)} \in \mathbb{R}^{N_I \times N_T}, \tag{4}$$

where $\mathcal{S}_L$ denotes the set of selected layers, H is the number of attention heads, and $\bar{\mathbf{A}}$ is the averaged attention map. Early denoising steps are preferred as they establish the coarse spatial layout of the edit.

To derive a per-image-token relevance score, we marginalize $\bar{\mathbf{A}}$ over the text token dimension:

$$\mathbf{w}_{\text{attn}}(i) = \sum_{j=1}^{N_T} \bar{A}(i,j), \quad i = 1, \ldots, N_I, \tag{5}$$

which measures the total attention each image token allocates to the editing instruction. The resulting vector $\mathbf{w} \in \mathbb{R}^{N_I}$ is reshaped to the latent spatial grid and normalized to unit mean to obtain the spatial weight matrix:

$$\mathbf{W} = \frac{\text{reshape}(\mathbf{w}_{\text{attn}}, H, W)}{\text{mean}(\mathbf{w}_{\text{attn}})} \in \mathbb{R}^{H \times W}, \tag{6}$$

where (H, W) is the spatial resolution of the latent representation. Thus, image regions with strong attention to the instruction tokens receive higher weights, while irrelevant regions are down-weighted.

Following DiffusionNFT [54], we adopt the NFT training objective that contrasts implicit positive and negative predictions. To incorporate spatial guidance, we reweight the loss at each location using $\mathbf{W}$:

$$\mathcal{L}^+ = \frac{1}{Z^+} \sum_{i,j} \mathbf{W}^{(i,j)} \cdot \|\hat{x}_0^+(i,j) - x_0(i,j)\|^2, \tag{7}$$

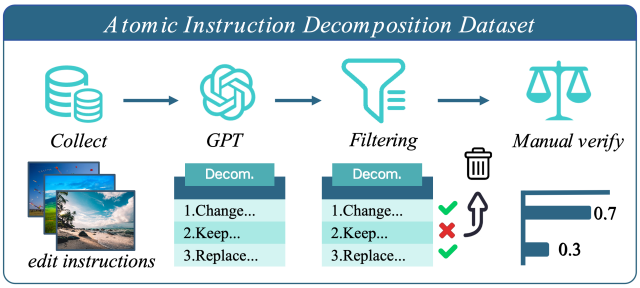

Figure 4: Pipeline for constructing the atomic instruction decomposition dataset.

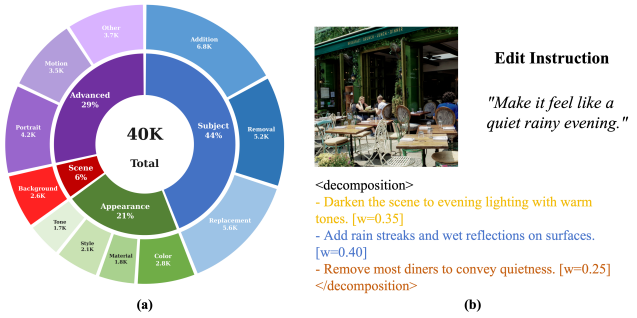

Figure 5: (a) Data distribution of atomic instruction decomposition dataset. (b) Example of instruction decomposition.

$$\mathcal{L}^- = \frac{1}{Z^-} \sum_{i,j} \mathbf{W}^{(i,j)} \cdot \|\hat{x}_0^-(i,j) - x_0(i,j)\|^2, \tag{8}$$

where $\hat{x}_0^+$ and $\hat{x}_0^-$ are implicit positive and negative predictions, respectively. The normalization factors Z^+ and Z^- are computed adaptively as:

$$Z^+ = \text{mean}\left(\left|\hat{x}_0^+ - x_0\right| \cdot \mathbf{W}\right), \quad Z^- = \text{mean}\left(\left|\hat{x}_0^- - x_0\right| \cdot \mathbf{W}\right), \tag{9}$$

which scale the loss based on the prediction error magnitude in instruction-relevant regions, preventing gradients from being dominated by easily predicted background areas. The final training loss is defined as:

$$\mathcal{L} = r \cdot \mathcal{L}^+ + (1 - r) \cdot \mathcal{L}^-, \tag{10}$$

where $r \in [0, 1]$ is the normalized reward. By concentrating gradient signals on regions the DiT itself identifies as instruction-relevant, this mechanism improves optimization efficiency and preserves fidelity in unedited areas.

4.4 Atomic Instruction Decomposition Dataset

To support the self-rewarding framework training, we employ a systematic workflow to construct the atomic instruction decomposition dataset. As illustrated in Figure 4, the process consists of three main stages. **(1) Sample Preparation.** As shown in Figure 5, we collect approximately 40K image editing samples across multiple editing categories, from ImgEdit [48], StyleBooth [14], and PromptFixData [50]. **(2) Automated Decomposition and Filtering.** Each instruction is automatically decomposed into atomic sub-instructions using GPT-5.2. To maintain focus on actionable edits,

Table 1: Quantitative comparison of state-of-the-art on ImgEdit-Bench across different editing tasks. "Overall" score is calculated as the average across all tasks.

Models	Add	Adjust	Extract	Replace	Remove	Background	Style	Hybrid	Action	Overall↑
MagicBrush [51]	2.84	1.58	1.51	1.97	1.58	1.75	2.38	1.62	1.22	1.83
Instruct-P2P [2]	2.45	1.83	1.44	2.01	1.50	1.44	3.55	1.20	1.46	1.88
AnyEdit [49]	3.18	2.95	1.88	2.47	2.23	2.24	2.85	1.56	2.65	2.45
UltraEdit [53]	3.44	2.81	2.13	2.96	1.45	2.83	3.76	1.91	2.98	2.70
ICEdit [52]	3.58	3.39	1.73	3.15	2.93	3.08	3.84	2.04	3.68	3.05
Step1X-Edit v1.1 [25]	3.88	3.14	1.76	3.40	2.41	3.16	4.63	2.64	2.52	3.06
UniWorld-V1 [23]	3.82	3.64	2.27	3.47	3.24	2.99	4.21	2.96	2.74	3.26
BAGEL [9]	3.81	3.59	1.58	3.85	3.16	3.39	4.51	2.67	4.25	3.42
OmniGen2 [42]	3.57	3.06	1.77	3.74	3.20	3.57	4.81	2.52	4.68	3.44
Flux-Kontext-dev [20]	3.83	3.65	2.27	4.45	3.17	3.98	4.55	3.35	4.29	3.71
Ovis-U1 [39]	3.99	3.73	2.66	4.38	4.15	4.05	4.86	3.43	4.68	3.97
GPT Image 1 [High] [31]	**4.61**	4.33	2.90	4.35	3.66	**4.57**	4.93	**3.96**	4.89	4.20
UniWorld-V2 [22]	4.29	4.44	4.32	4.69	4.72	4.41	4.91	3.83	4.83	4.49
Qwen-Image-Edit [41]	4.38	4.16	3.43	4.66	4.14	4.38	4.81	3.82	4.69	4.27
SREdit (Ours)	4.45	**4.53**	**4.41**	**4.75**	**4.78**	4.48	**4.95**	3.88	**4.90**	**4.60**

Table 2: Quantitative evaluation of state-of-the-art image editing methods on GEdit-Bench and KRIS-Bench

Model	GEdit-Bench			KRIS-Bench			
	SC	PQ	OS↑	FK	CK	PK	OS↑
ICEdit [52]	4.94	7.39	4.87	46.99	42.73	27.76	40.70
Omnigen [45]	5.88	5.87	5.01	33.11	28.02	23.89	28.85
Omnigen 2 [42]	7.16	6.77	6.41	57.36	44.20	47.79	49.71
BAGEL-thinking [9]	7.70	6.51	6.66	66.18	61.92	49.02	60.18
BAGEL [9]	7.48	6.80	6.60	60.26	55.86	51.69	56.21
Uniworld-V1 [23]	4.93	7.43	4.85	47.71	44.80	47.92	50.27
Hidream-I1 [3]	5.66	6.06	5.01	43.31	50.05	37.64	44.72
Hidream-E1.1 [3]	7.15	6.65	6.42	43.52	44.71	36.08	42.25
Flux-Kontext-dev [20]	7.16	7.37	6.51	53.28	50.36	42.53	49.54
UniWorld-Kontext [22]	7.28	7.49	6.74	55.50	51.39	43.76	51.04
Step1X-Edit v1.1 [25]	7.66	7.35	6.97	53.05	54.34	44.66	51.59
Qwen-Image-Edit [41]	8.00	7.86	7.56	61.47	56.79	47.07	56.15
DiffusionNFT [54]	8.04	7.63	7.77	62.38	58.21	48.56	57.89
SREdit (Ours)	8.47	8.27	**8.07**	65.12	61.35	52.18	**62.47**

we apply rule-based filtering to remove non-editing directives (e.g., "Preserve the background unchanged"). **(3) Expert Annotation.** The filtered data is rigorously reviewed by professional annotators. For each sample, they ensure that 1) every sub-instruction is reasonable and well-defined, and 2) no sub-instruction contradicts the main editing instruction. Annotators also assign importance weights to each sub-instruction, providing structured, high-quality supervision for instruction decomposition and reward modeling.

5 Experiments

5.1 Experimental Setup

5.1.1 Benchmarks. We conduct experiments on three widely-used image editing benchmarks. GEdit-Bench and ImgEdit-Bench evaluate fundamental image editing capabilities, while KRIS-Bench [44] targets advanced reasoning and abstract instruction interpretation. On GEdit-Bench, we report Semantic Consistency (SC), Perceptual Quality (PQ), and Overall Score (OS) using VIEScore [19], which is evaluated by GPT-4.1 [32]. On ImgEdit-Bench, we also use GPT-4.1 to rate each sample from 1 to 5 based on instruction adherence, editing quality, and detail preservation, with the final score capped by instruction adherence. On KRIS-Bench, we use GPT-4o to provide 1-5 ratings for Visual Consistency, Visual Quality, Instruction Following, and Knowledge Plausibility.

5.1.2 Baselines. We compare our method against state-of-the-art image editing models, including ICEdit [52], Omnigen [45], Omnigen 2 [42], BAGEL [9], Uniworld [23], Flux-Kontext-Dev [20], Step1X-Edit [25], Qwen-Image-Edit-2509 [41], DiffusionNFT [54] (finetuned on Qwen-Image-Edit-2509).

5.1.3 Implementation Details. We build SREdit upon Qwen-Image-Edit and use EditScore [26] as the base reward MLLM. We conduct two-stage training. In stage one, we finetune the MLLM on our instruction decomposition dataset, to jointly output decomposed instructions with their importance weights. In stage two, we finetune the DiT using RL under the self-reward framework. During RL, we sample 12 images per prompt and train same steps for our methods as DiffusionNFT. Experiments are conducted on eight NVIDIA H20 GPUs. More training details are in the supplementary materials.

5.2 Results

5.2.1 Quantitative Results. As shown in Table 2 and Table 1, SREdit achieves state-of-the-art performance across three image editing benchmarks. Compared to existing methods, it improves overall scores by 3.8% (7.77 to 8.07) on GEdit-Bench, 2.5% (4.49 to 4.60) on ImgEdit-Bench, and 3.8% (60.18 to 62.47) on KRIS-Bench.

On GEdit-Bench and ImgEdit-Bench, which evaluate fundamental editing capabilities, SREdit outperforms all open-source models. Notably, compared to the base model Qwen-Image-Edit and its RL-enhanced variant DiffusionNFT, SREdit demonstrates consistent improvements in both semantic consistency and perceptual quality, validating the effectiveness of our self-reward framework in providing more aligned optimization signals.

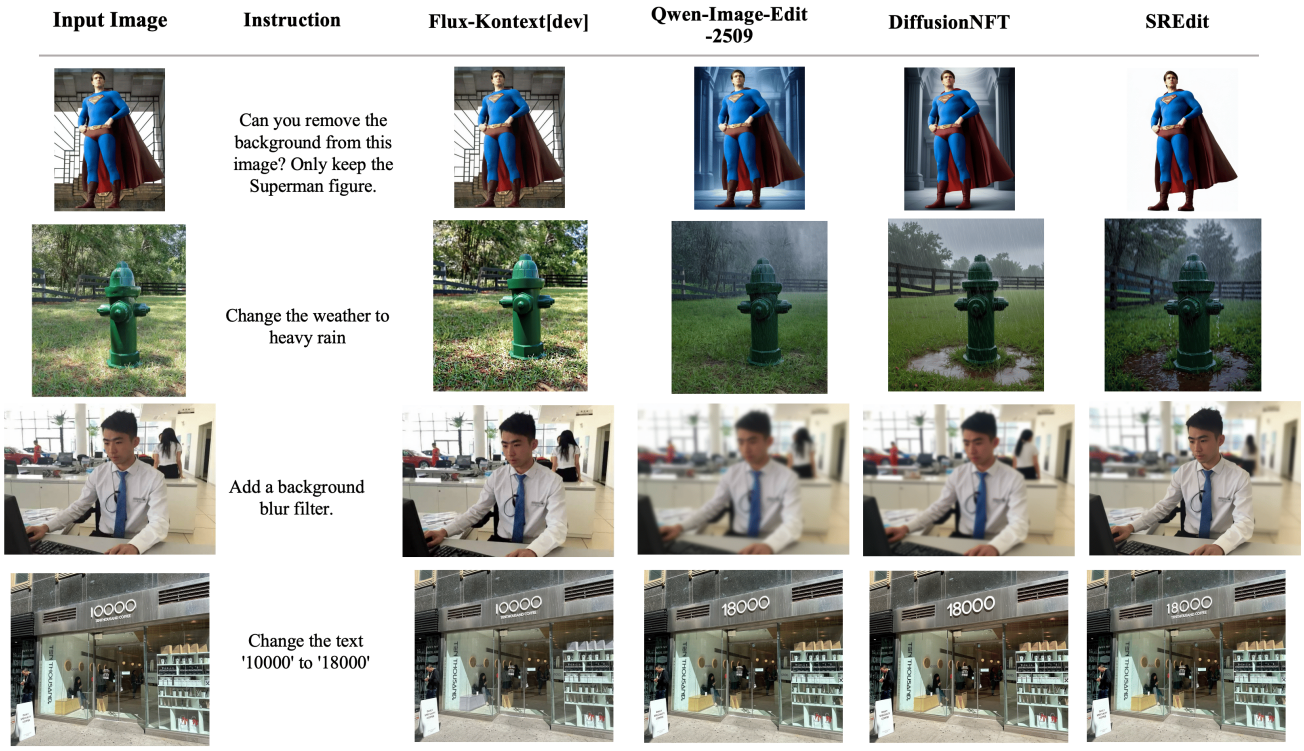

Figure 6: Qualitative comparison of SREdit with previous methods.

On the more challenging KRIS-Bench, which requires advanced reasoning and abstract instruction interpretation, SREdit achieves an overall score of 62.47, outperforming Qwen-Image-Edit (56.15). The performance gains suggest that our instruction decomposition mechanism effectively breaks down complex instructions into atomic sub-instructions, enabling the model to better reason and execute multi-step editing operations.

5.2.2 Qualitative Results. Figure 6 presents qualitative comparisons, highlighting three key advantages of SREdit. (1) **Superior instruction adherence**: In the first row, only SREdit successfully removes the background while preserving the Superman figure with clear boundaries. (2) **More realistic details**: In the second row, SREdit generates more detailed rain effects, benefiting from our instruction decomposition that provides fine-grained guidance. (3) **Region-aware editing**: In rows 3 and 4, SREdit precisely modifies only target regions—applying background blur while keeping the foreground person sharp, and editing text from "10000" to "18000" without altering surrounding elements. This demonstrates the effectiveness of our region-aware reweighting mechanism.

5.2.3 User Study. In addition to GPT-based evaluation, we conducted a human study with professional annotators. We randomly sampled 200 editing tasks from GEdit-Bench and compared five methods: Flux-Kontext-dev, BAGEL, Qwen-Image-Edit, DiffusionNFT, and SREdit (Ours). Ten professional annotators with backgrounds in image editing were recruited for this evaluation. For each test case, annotators were provided with the source image, the editing instruction, and the edited results, which were presented in randomized order without revealing method labels. Each result was rated on three criteria using a score of 1 to 5: (1) **Instruction Following**: How faithfully the edited image reflects all aspects of the editing instruction. (2) **Visual Quality**: The overall perceptual quality, naturalness, and absence of artifacts. (3)**Edit Precision**: Whether modifications are accurately localized to the intended regions, leaving irrelevant areas unchanged. The results, summarized in Figure 7, demonstrate that the evaluators prefer SREdit in all three dimensions.

Table 3: Ablation studies of SREdit on GEdit-Bench.

Method	SC	PQ	Overall Score
Qwen-Image-Edit (baseline)	8.00	7.86	7.56
+ Aligned Reasoning	8.24	8.08	7.79
+ Unified Model	8.41	8.15	7.90
+ Spatial Reward Reweighting	**8.47**	**8.27**	**8.07**

5.3 Ablation Studies and Analysis

5.3.1 Effectiveness of Each Component. To evaluate the effectiveness of each component, we conduct ablation studies by progressively adding components to the baseline. The results are shown in Table 3, evaluating on GEdit-Bench. **(1) Effect of aligned reasoning.** We first evaluate the aligned reasoning mechanism, where decomposed sub-instructions are explicitly shared with the reward

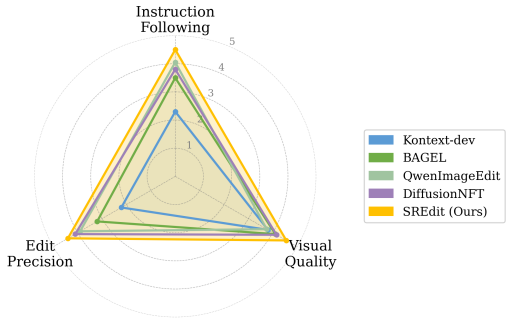

Figure 7: User study results: SREdit receives the highest human evaluation scores across all dimensions.

model. This approach improves the overall score by 0.23, demonstrating that aligning fine-grained instruction decomposition with reward evaluation effectively reduces semantic inconsistency. **(2) Effect of unified model.** We further unify instruction understanding and reward evaluation within the same MLLM. This design improves the overall score by 0.11, by mitigating model inter-model bias. **(3) Effect of region-aware reward reweighting.** Finally, incorporating the region-aware reward reweighting mechanism further improves the overall score by 0.17. This confirms that spatially-weighted supervision directs the optimization toward specific regions related with instructions, enabling precise learning.

5.3.2 Visualization of Region-Aware Attention. We visualize the attention maps extracted from DiT during evaluation. As shown in Figure 8, for an editing instruction, we display the source image, edited image and corresponding attention heatmaps. We observe that the attention maps accurately highlight the regions on the face mask, which allows more accurate learning in the RL training.

Figure 8: Visualization of reward model attention map.

5.3.3 Illustration of Instruction Decomposition. Figure 9 illustrates how instruction decomposition enhances editing quality. Given the abstract instruction "Modify it into an Impressionist oil painting," our MLLM decomposes it into three concrete sub-instructions covering brush stroke texture, edge softening and painterly motion. The model without decomposition produces a result that fails to capture fine-grained painterly details, whereas our decomposition-guided result faithfully reproduces Impressionist characteristics. This demonstrates that explicit atomic decomposition provides more precise optimization targets, leading to both higher instruction compliance and superior visual fidelity.

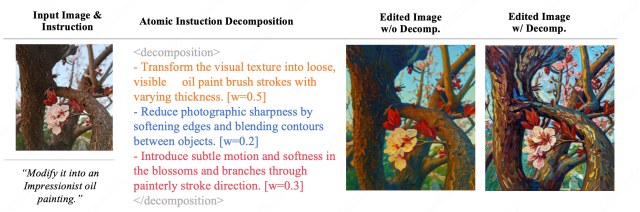

Figure 9: An instruction decomposition example.

5.4 Generalization to Text-to-Image Generation

To verify the generalizability of our self-rewarding paradigm beyond image editing, we further extend the framework to the text-to-image (T2I) generation task, which we denote as SREdit-T2I. Specifically, we employ Qwen-Image [41] as the base generative model and conduct training on the FLUX-Reason-6M [11] dataset. Our training pipeline follows the Self-Rewarding framework described in Section 4, consisting of two main phases: atomic instruction fine-tuning of the MLLM and RL-based training for the DiT. For the reward model, we utilize HPSv3 [27] to evaluate human preference. We conduct evaluations on the OneIG-Bench(ZH) [4] and compare our approach with current SOTA T2I methods. As shown in Table 4, our method achieves an overall score of 0.563, surpassing the baseline by 2.7%. These results demonstrate that the self-rewarding decomposition mechanism consistently brings performance improvements across different generative tasks.

Table 4: Performance comparison of Text-to-Image methods on OneIG-Bench(ZH).

Model	Alignment	Text	Reasoning	Style	Diversity	Overall↑
Janus-Pro [7]	0.324	0.148	0.104	0.264	**0.358**	0.240
BLIP3-o [5]	0.608	0.092	0.213	0.369	0.233	0.303
BAGEL [9]	0.672	0.365	0.186	0.357	0.268	0.370
BAGEL+CoT [9]	0.719	0.127	0.219	0.385	0.197	0.329
Cogview4 [55]	0.700	0.193	0.236	0.348	0.214	0.338
Lumina-Image 2.0 [34]	0.731	0.136	0.221	0.343	0.240	0.334
HiDream-I1-Full [3]	0.620	0.205	0.256	0.304	0.300	0.337
Kolors 2.0 [18]	0.738	0.502	0.226	0.331	0.333	0.426
Seedream 3.0 [12]	0.793	0.928	0.281	0.397	0.243	0.528
GPT Image 1 [High] [31]	0.812	0.650	**0.300**	**0.449**	0.159	0.474
Qwen-Image [41]	0.825	0.963	0.267	0.405	0.279	0.548
SREdit-T2I (Ours)	**0.831**	**0.968**	0.285	0.419	0.271	**0.563**

6 Conclusion

In this work, we identify a cognitive mismatch problem in RL-based image editing, where MLLM's instruction-understanding may conflict with the evaluation criteria of an external reward model. To resolve this, we propose a novel framework SREdit. Our method decomposes instructions into atomic sub-instructions to enable explicit task sharing, utilizes the same MLLM for self-consistent optimization, and incorporates a spatial reward reweighting strategy to focus learning on semantically relevant regions. Experiments conducted on GEdit-Bench, KRIS-Bench, and ImgEdit-Bench demonstrate that SREdit consistently outperforms strong baselines across all benchmarks.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. 2025. Qwen2.5-VL Technical Report. arXiv:2502.13923 [cs.CV] https://arxiv.org/abs/2502.13923

[2] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. 2023. InstructPix2Pix: Learning to Follow Image Editing Instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 18392–18402.

[3] Qi Cai, Jingwen Chen, Yang Chen, Yehao Li, Fuchen Long, et al. 2025. HiDream-I1: A High-Efficient Image Generative Foundation Model with Sparse Diffusion Transformer. *arXiv preprint arXiv:2505.22705* (2025).

[4] Jingjing Chang, Yixiao Fang, Peng Xing, Shuhan Wu, Wei Cheng, et al. 2025. OneIG-Bench: Omni-dimensional Nuanced Evaluation for Image Generation. *arXiv preprint arXiv:2506.07977* (2025).

[5] Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, et al. 2025. BLIP3-o: A Family of Fully Open Unified Multimodal Models-Architecture, Training and Dataset. *arXiv preprint arXiv:2505.09568* (2025).

[6] Sherry X. Chen, Misha Sra, and Pradeep Sen. 2025. Instruct-CLIP: Improving Instruction-Guided Image Editing with Automated Data Refinement Using Contrastive Learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 28513–28522.

[7] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, et al. 2025. Janus-Pro: Unified Multimodal Understanding and Generation with Data and Model Scaling. *arXiv preprint arXiv:2501.17811* (2025).

[8] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. 2022. DiffEdit: Diffusion-based semantic image editing with mask guidance. In *The Eleventh International Conference on Learning Representations*.

[9] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, et al. 2025. Emerging Properties in Unified Multimodal Pretraining. *arXiv preprint arXiv:2505.14683* (2025).

[10] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, et al. 2023. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems* 36 (2023), 79858–79885.

[11] Rongyao Fang, Aldrich Yu, Chengqi Duan, Linjiang Huang, Shuai Bai, et al. 2025. FLUX-Reason-6M & PRISM-Bench: A Million-Scale Text-to-Image Reasoning Dataset and Comprehensive Benchmark. *arXiv preprint arXiv:2509.09680* (2025).

[12] Yu Gao, Lixue Gong, Qiushan Guo, Xiaoxia Hou, Zhichao Lai, et al. 2025. Seedream 3.0 Technical Report. *arXiv preprint arXiv:2504.11346* (2025).

[13] Google. 2025. Gemini 3 Flash. https://blog.google/technology/google-deepmind/gemini-3-flash/.

[14] Zhen Han, Chaojie Mao, Zeyinzi Jiang, Yulin Pan, and Jingfeng Zhang. 2025. StyleBooth: Image Style Editing with Multimodal Instruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1947–1957.

[15] Huiguo He, Tianfu Wang, Huan Yang, Jianlong Fu, Nicholas Jing Yuan, et al. 2023. Learning Profitable NFT Image Diffusions via Multiple Visual-Policy Guided Reinforcement Learning. In *Proceedings of the 31st ACM International Conference on Multimedia*. 6831–6840.

[16] Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, et al. 2024. HQ-Edit: A High-Quality Dataset for Instruction-based Image Editing. *arXiv preprint arXiv:2404.09990* (2024).

[17] Siyu Jiao, Yiheng Lin, Yujie Zhong, Qi She, Wei Zhou, et al. 2025. ThinkGen: Generalized Thinking for Visual Generation. *arXiv preprint arXiv:2512.23568* (2025).

[18] Kolors Team. 2024. Kolors: Effective Training of Diffusion Model for Photorealistic Text-to-Image Synthesis. *Technical Report* (2024). https://github.com/Kwai-Kolors/Kolors.

[19] Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. 2024. VIEScore: Towards Explainable Metrics for Conditional Image Synthesis Evaluation. *arXiv preprint arXiv:2312.14867* (2024).

[20] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, et al. 2025. FLUX.1 Kontext: Flow Matching for In-Context Image Generation and Editing in Latent Space. *arXiv preprint arXiv:2506.15742* (2025).

[21] Seung Hyun Lee, Yinxiao Li, Junjie Ke, Innfarn Yoo, Han Zhang, et al. 2024. Parrot: Pareto-Optimal Multi-reward Reinforcement Learning Framework for Text-to-Image Generation. In *Proceedings of the European Conference on Computer Vision*. 462–478.

[22] Zongjian Li, Zheyuan Liu, Qihui Zhang, Bin Lin, Feize Wu, et al. 2025. Uniworld-V2: Reinforce Image Editing with Diffusion Negative-aware Finetuning and MLLM Implicit Feedback. *arXiv preprint arXiv:2510.16888* (2025).

[23] Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, et al. 2025. UniWorld-V1: High-Resolution Semantic Encoders for Unified Visual Understanding and Generation. *arXiv preprint arXiv:2506.03147* (2025).

[24] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, et al. 2025. Flow-GRPO: Training Flow Matching Models via Online RL. *arXiv preprint arXiv:2505.05470* (2025).

[25] Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, et al. 2025. Step1X-Edit: A Practical Framework for General Image Editing. *arXiv preprint arXiv:2504.17761* (2025).

[26] Xin Luo, Jiahao Wang, Chenyuan Wu, Shitao Xiao, Xiyan Jiang, et al. 2025. EditScore: Unlocking Online RL for Image Editing via High-Fidelity Reward Modeling. *arXiv preprint arXiv:2509.23909* (2025).

[27] Yuhang Ma, Yunhao Shui, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. 2025. HPSv3: Towards Wide-Spectrum Human Preference Score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 15086–15095.

[28] Qi Mao, Lan Chen, Yuchao Gu, Zhen Fang, and Mike Zheng Shou. 2024. MAG-Edit: Localized Image Editing in Complex Scenarios via Mask-Based Attention-Adjusted Guidance. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 6842–6850.

[29] OpenAI. 2024. GPT-4o System Card. *arXiv preprint arXiv:2410.21276* (2024).

[30] OpenAI. 2025. GPT-5.2. https://help.openai.com/en/articles/9624314-model-release-notes.

[31] OpenAI. 2025. Introducing 4o image generation. https://openai.com/index/introducing-4o-image-generation/.

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

[33] Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, et al. 2025. Transfer between Modalities with MetaQueries. *arXiv preprint arXiv:2504.06256* (2025).

[34] Qi Qin, Le Zhuo, Yi Xin, Ruoyi Du, Zhen Li, et al. 2025. Lumina-Image 2.0: A Unified and Efficient Image Generative Framework. In *Proceedings of the IEEE/CVF international conference on computer vision*. 20031–20042.

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, et al. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning*. PMLR, 8748–8763.

[36] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, et al. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300* (2024).

[37] Joonghyuk Shin, Alchan Hwang, Yujin Kim, Daneul Kim, and Jaesik Park. 2025. Exploring Multimodal Diffusion Transformers for Enhanced Prompt-based Image Editing. In *Proceedings of the IEEE/CVF international conference on computer vision*. 19492–19502.

[38] Xueyun Tian, Wei Li, Bingbing Xu, Yige Yuan, Yuanzhuo Wang, et al. 2025. MIGE: Mutually Enhanced Multimodal Instruction-Based Image Generation and Editing. In *Proceedings of the 33rd ACM International Conference on Multimedia*. 10622–10631.

[39] Guo-Hua Wang, Shanshan Zhao, Xinjie Zhang, Liangfu Cao, Pengxin Zhan, et al. 2025. Ovis-U1 Technical Report. *arXiv preprint arXiv:2506.23044* (2025).

[40] Qian Wang, Aleksandar Cvejic, Abdelrahman Eldesokey, and Peter Wonka. 2025. EditCLIP: Representation Learning for Image Editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 15960–15970.

[41] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, et al. 2025. Qwen-Image Technical Report. *arXiv preprint arXiv:2508.02324* (2025).

[42] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, et al. 2025. OmniGen2: Exploration to Advanced Multimodal Generation. *arXiv preprint arXiv:2506.18871* (2025).

[43] Mingrui Wu, Lu Wang, Pu Zhao, Fangkai Yang, Jianjin Zhang, et al. 2025. Reprompt: Reasoning-augmented reprompting for text-to-image generation via reinforcement learning. *arXiv preprint arXiv:2505.17540* (2025).

[44] Yongliang Wu, Zonghui Li, Xinting Hu, Xinyu Ye, Xianfang Zeng, et al. 2025. KRIS-Bench: Benchmarking Next-Level Intelligent Image Editing Models. *arXiv preprint arXiv:2505.16707* (2025).

[45] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, et al. 2025. OmniGen: Unified Image Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13294–13304.

[46] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, et al. 2023. ImageReward: Learning and Evaluating Human Preferences for Text-to-Image Generation. *Advances in Neural Information Processing Systems* 36 (2023), 15903–15935.

[47] Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, et al. 2025. DanceGRPO: Unleashing GRPO on Visual Generation. *arXiv preprint arXiv:2505.07818* (2025).

[48] Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, et al. 2025. ImgEdit: A Unified Image Editing Dataset and Benchmark. *arXiv preprint arXiv:2505.20275* (2025).

[49] Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, et al. 2025. AnyEdit: Mastering Unified High-Quality Image Editing for Any Idea. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 26125–26135.

[50] Yongsheng Yu, Ziyun Zeng, Hang Hua, Jianlong Fu, and Jiebo Luo. 2024. PromptFix: You Prompt and We Fix the Photo. *arXiv preprint arXiv:2405.16785* (2024).

[51] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. 2023. MagicBrush: A Manually Annotated Dataset for Instruction-Guided Image Editing. *Advances in Neural Information Processing Systems* 36 (2023), 31428–31449.

[52] Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. 2025. In-Context Edit: Enabling Instructional Image Editing with In-Context Generation in Large

Scale Diffusion Transformer. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[53] Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, et al. 2024. UltraEdit: Instruction-based Fine-Grained Image Editing at Scale. *Advances in Neural Information Processing Systems* 37 (2024), 3058–3093.

[54] Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, et al. 2025. DiffusionNFT: Online Diffusion Reinforcement with Forward Process. *arXiv preprint arXiv:2509.16117* (2025).

[55] Wendi Zheng, Jiayan Teng, Zhuoyi Yang, Weihan Wang, Jidong Chen, Xiaotao Gu, Yuxiao Dong, Ming Ding, and Jie Tang. 2024. CogView3: Finer and Faster Text-to-Image Generation via Relay Diffusion. *arXiv preprint arXiv:2403.05121* (2024).

[56] ZhipuAI. 2025. GLM-Image. https://z.ai/blog/glm-image

[57] Chao Zhou, Tianyi Wei, and Nenghai Yu. 2025. Scale Your Instructions: Enhance the Instruction-Following Fidelity of Unified Image Generation Model by Self-Adaptive Attention Scaling. In *Proceedings of the IEEE/CVF international conference on computer vision*. 15171–15181.

[58] Jun Zhou, Jiahao Li, Zunnan Xu, Hanhui Li, Yiji Cheng, et al. 2025. FireEdit: Fine-grained Instruction-based Image Editing via Region-aware Vision Language Model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 13093–13103.