CAN WE PERFORM ONLINE RL FOR IMAGE EDITING WITHOUT EDITING REWARDS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) enables direct preference optimization for image editing through editing-specific rewards, which remain less developed due to costly 'edit instruction, source & edited image' triplet supervision and complex task-dependent calibration. In contrast, text-to-image (T2I) generation benefits from a mature and diverse reward ecosystem spanning semantic alignment, aesthetics, realism, glyph rendering, and other visual preferences. Extending this ecosystem to image editing would substantially broaden the range of visual preferences for RL-based optimization, raising the natural question: *Can We Perform Image Editing RL without Editing Rewards?* In this paper, we argue that the standard image editing dimensions have potential to be mapped to the T2I reward space: image quality can transfer directly, prompt following can be aligned through a description of the desired visual state, and reference consistency admits a coarse semantic conversion by encoding the source content to preserve. However, building such lever is non-trivial. Editing instructions specify relative changes, whereas T2I rewards require self-contained target descriptions; moreover, semantically correct captions to humans may not induce the frozen reward to rank successful edits above failures. We therefore introduce *Lever-Edit*, a two-stage reward-transfer framework. Stage 1 learns a textual reward interface that produces both semantically faithful and reward-aligned descriptions of desired post-edit states with a pluggable target-state adapter, change-aware source–target contrast, two-level preference calibration, and regularizations. Stage 2 freezes this interface and optimizes editing policy solely with the transferred T2I reward. Experiments show competitive editing alignment and source preservation against editing-reward-based fine-tuning, while outperforming intuitive transfer baselines across various benchmarks.

1 INTRODUCTION

Instruction-guided image editing enables users to modify an existing image through natural-language instructions while preserving content unrelated to the requested change (Huang et al., 2025). Although recent diffusion and flow-based editors have achieved substantial progress (Nichol et al., 2021; Liu et al., 2025b; Yan et al., 2026b), supervised fine-tuning optimizes likelihood over curated editing pairs rather than task-level human preferences. Reinforcement learning (RL) instead directly optimizes scalar or rationale-based feedback on sampled results (Black et al., 2023; Fan et al., 2023; Guo et al., 2026; Liu et al., 2026a). At the center of editing RL lies the reward. However, constructing a reliable editing reward requires joint reasoning over the reference image, editing instruction, and edited output, as well as costly triplet-level supervision. The desired balance between executing the instruction and preserving source content further varies across edit types, complicating reward calibration and generalization (Wu et al., 2025b; Luo et al., 2026; Zhao et al., 2026; Long et al., 2026). Editing quality is commonly characterized along three dimensions: *pure image quality*, *edit execution*, and *reference consistency* (Luo et al., 2026; Zhao et al., 2026). Detailed related work can be found in Appx. A. Pure image quality concerns intrinsic properties of the edited output, edit execution measures whether the requested transformation is realized, and reference consistency measures the preservation of source content unrelated to the edit. Surprisingly, we find that all three dimensions exhibit potential mappings to the T2I reward space: image quality can be evaluated directly, edit execution can be reformulated as alignment with a description of the

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

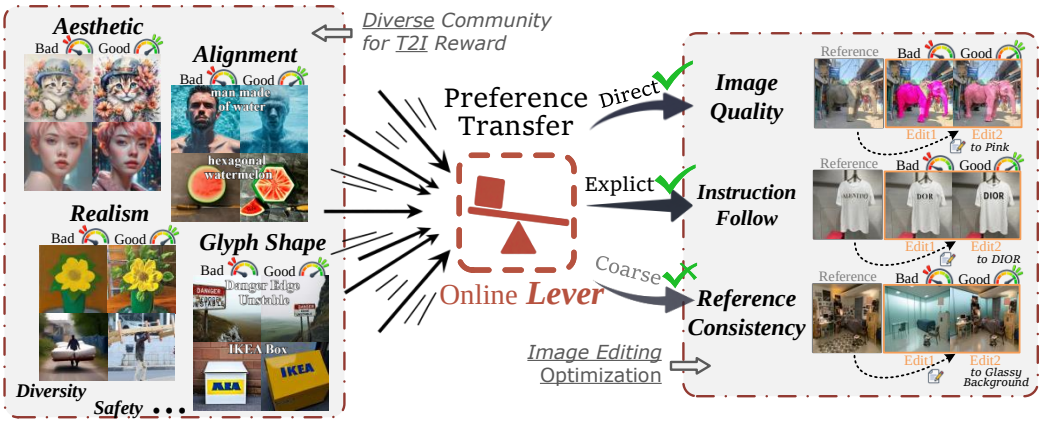


Figure 1: Motivation for T2I-to-editing reward transfer. *Lever-Edit* acts as an online lever that harnesses the collective power of the broad T2I reward community, transferring its mature and diverse visual preferences into effective supervision for image-editing RL.

desired visual state, and reference consistency can be coarsely projected by incorporating preserved source semantics into that description, as illustrated in Fig. 1.

This raises a central question: *Can We Perform Image Editing RL without Editing Rewards*, instead using a T2I reward? Such a formulation offers two advantages. First, it gives editing RL direct access to a mature and continuously expanding T2I reward ecosystem, spanning VLM-based evaluators for semantic alignment, aesthetics, realism, glyph (OCR), safety, and other visual preferences (Xu et al., 2023; Wu et al., 2023b; Kirstain et al., 2023; Ye et al., 2025; Zhu et al., 2026; Wang et al., 2025; Lamba et al., 2025), as well as recent zero-shot and annotation-free reward formulations (Liu et al., 2026b; Huang et al., 2026). Second, it avoids the costly design, annotation, development, and deployment of native editing evaluators. However, transferring T2I rewards to image editing is nontrivial, requiring descriptions that **1)** faithfully represent the target state, **2)** induce the correct editing preference, **3)** and reduce reward hacking.

In this paper, we introduce *Lever-Edit*, a reward-transfer framework in which a lightweight reward-aligned captioner acts as a ***lever***: a module that enables mature T2I reward to provide effective signal for image-editing RL. Specifically, *Lever-Edit* follows a two-stage design. In the first stage, we adapt a VLM through prefix tuning to map a reference image and an editing instruction to a counterfactual target caption that combines the desired change with the source semantics to preserve. We further align the lever's representation with the frozen T2I reward, producing reward-compatible conditions rather than generic image descriptions. To make this representation effective for editing, a change-aware reward emphasizes edit-relevant information over shared content, while a preference-aware reward treats the source image as a self-supervised negative under the target caption, thereby enforcing the desired preference direction. Moreover, semantic and distributional anchors regularize the captioner against reward-specific drift. In the second stage, we freeze the captioner and optimize the diffusion editing policy using only the transferred T2I reward , without invoking any editing reward. By separating reward transfer from policy optimization, *Lever-Edit* is pluggable with different VLMs, T2I rewards, and editing backbones.

Our contributions are summarized as follows:

- We argue that image-editing RL can leverage the mature T2I reward space through reward transfer, offering an alternative to dedicated editing-specific reward models while retaining strong potential to improve editing generation quality.
- We propose *Lever-Edit*, a two-stage framework that decouples reward transfer from policy optimization, enabling effective transfer of T2I rewards to image editing with flexible compatibility across VLMs, reward models, and editing backbones.
- Extensive reward- and policy-level evaluations demonstrate the effectiveness of T2I reward transfer for image editing, with *Lever-Edit* consistently outperforming direct-transfer baselines and remaining competitive with specialized editing rewards.

2 PROBLEM FORMULATION

2.1 REWARD-TRANSFER SETUP

Let I_{ref} denote a reference image, q_{edit} an editing instruction, and $I_{\text{edit}} \sim \pi_\theta(\cdot \mid I_{\text{ref}}, q_{\text{edit}})$ the output of an editing policy. Conventional editing RL optimizes a task-specific reward over the complete editing triplet:

$$\max_\theta \ \mathbb{E}_{I_{\text{edit}} \sim \pi_\theta(\cdot \mid I_{\text{ref}}, q_{\text{edit}})} \left[R_{\text{edit}}(I_{\text{ref}}, q_{\text{edit}}, I_{\text{edit}}) \right]. \tag{1}$$

Because R_{edit} observes the complete triplet, it can assess both whether the requested change is executed and whether unrelated source content is preserved. A T2I reward instead evaluates an image–text pair $R_{\text{T2I}}(I, c)$, where c is expected to describe the visual content of I as a self-contained target state. Directly substituting q_{edit} for c therefore creates a conditioning mismatch: an editing instruction specifies a relative transformation whose desired outcome depends on I_{ref}, but does not describe the complete edited image or the source content that should remain unchanged.

2.2 MAPPING EDITING QUALITY TO THE T2I REWARD SPACE

Despite this interface mismatch, the standard dimensions of editing quality admit semantic correspondences in the T2I reward space. Editing quality is commonly characterized by *pure image quality*, *edit execution*, and *reference consistency* (Luo et al., 2026; Zhao et al., 2026), which we consider separately without assuming a particular scalarization. Pure image quality transfers directly because it is intrinsic to I_{edit}. Edit execution can be reformulated by converting the relative instruction into an absolute description c_{change} of the intended post-edit state, thereby reducing execution assessment to image–text alignment. Reference consistency is less direct because it concerns the relation between I_{ref} and I_{edit}. Nevertheless, the source semantics unrelated to the requested edit can be represented by a complementary description c_{preserve}.

These correspondences define a condition mapping:

$$c_{\text{tgt}} = g(I_{\text{ref}}, q_{\text{edit}}) = c_{\text{change}} \oplus c_{\text{preserve}}, \tag{2}$$

where $\oplus$ denotes the semantic composition of the intended modification and the source content to preserve. The resulting target condition supports the transferred objective:

$$\max_\theta \ \mathbb{E}_{I_{\text{edit}} \sim \pi_\theta} \left[R_{\text{T2I}} \left(I_{\text{edit}}, c_{\text{tgt}} \right) \right]. \tag{3}$$

Therefore, pure image quality remains directly assessable from the edited output, while edit execution and reference consistency are projected into alignment with a unified target state.

Admittedly, a slight gap exists between T2I and image-pair evaluation. Textual conditions can represent objects, attributes, and relations that should change or unchanged, but cannot fully specify low-level pixel-accurate preservation. Instead, the editing process inherently handles pixel-level preservation. Therefore, such formulation establishes the semantic transferability of the standard editing objectives while leaving open how the target condition $g(I_{\text{ref}}, q_{\text{edit}})$ should be constructed.

2.3 INTUITIVE REWARD-TRANSFER PROTOCOLS

Eq. 2 suggests several straightforward transfer protocols. We introduce three representative variants that differ in how they instantiate g.

Direct Edit Prompt. *(Lever-Pure-Edit)* sets $c_{\text{tgt}} = q_{\text{edit}}$, avoids conversion costs, but treats a relative instruction completely as description of edited image. This results in source information loss. **Online VLM Rewriting.** *(Lever-Pure-VLM)* instead employs a general-purpose VLM to generate the target condition given source image and instruction as inputs: $c_{\text{tgt}} = g_{\text{VLM}}(I_{\text{ref}}, q_{\text{edit}})$. This explicitly preserves more source contents, but repeated VLM inference introduces computation overhead. Moreover, generated outputs are not calibrated to the representation space of T2I reward, therefore cannot provide reliable signals for image editing alignment. **Offline Proprietary-MLLM Rewriting.** *(Lever-Offline-SOTA).* A strong proprietary MLLM can precompute target descriptions offline, using its instruction-following and visual-reasoning capabilities to capture both intended edit and source content to preserve. However, these descriptions are generated independently of the

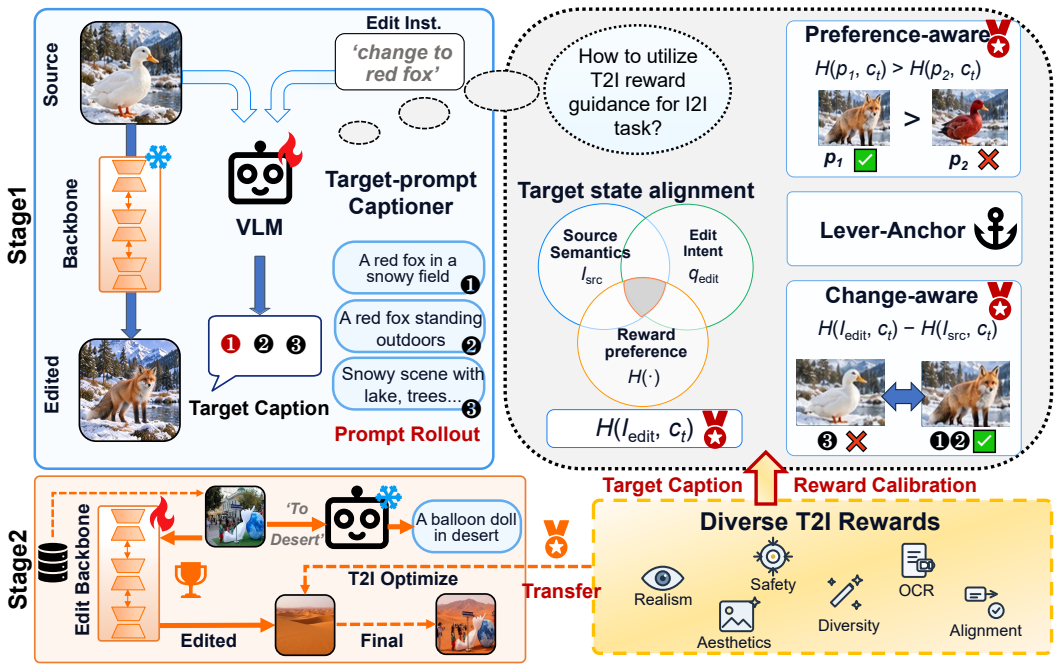


Figure 2: Overall two-stage *Lever-Edit* T2I reward transfer pipeline. In Stage 1, *Lever-Edit* finetunes a VLM captioner for edited state description with edit backbone frozen. It utilize target state alignment, change-aware and preference-aware reward feedback to utilize T2I preferences for effective edit signals. In Stage 2, VLM is frozen and gets T2I signals as edit rewards for finetuning backbones.

downstream T2I reward, making them inflexible across reward models and optimization needs while introducing dependence on a closed model. More fundamentally, semantic accuracy to humans does not ensure reward compatibility: a description may emphasize attributes weakly represented by the reward or fall outside its preferred conditioning distribution.

Together, these protocols reveal two requirements for effective reward transfer: the target description must faithfully encode both the intended change and the source semantics to preserve, while remaining compatible with the frozen T2I reward. The direct edit prompt fails to specify a complete target state, whereas generic VLM or MLLM rewriting does not guarantee reward-specific alignment.

3 Proposed Method

3.1 Lever Training Challenges and Pipeline Overview

We train the Lever as a target-description policy g_ϕ, optimized directly from feedback provided by frozen T2I rewards. However, learning g_ϕ through reward-driven prompt optimization introduces two further risks. **First**, the VLM may exploit reward-preferred words or stylistic patterns that increase the T2I reward without faithfully describing the edited image. **Second**, because the source and edited images share substantial content, the generated prompt may overemphasize common details that improve overall image–text alignment but provide little edit-specific supervision. Overall, effective target-prompt conversion must address three coupled challenges: constructing a complete and faithful target state, aligning it with the frozen reward to obtain an edit-discriminative signal, and preventing reward-driven optimization from exploiting reward-specific shortcuts.

Therefore, **Lever Calibration** addresses first two requirements by constructing editing-aware reward feedback, while **Lever Anchoring** constrains semantic and distributional drift. As in Fig. 2, we learn the Lever as a target-description policy g_ϕ that converts an editing input into a reward-compatible target description. Stage 1 adapts only the parameter-efficient VLM parameters: Lever Calibration provides target-state, change-aware, and preference supervision, while Lever Anchor-

ing regularizes semantic and distributional drift. Stage 2 freezes the learned Lever and uses target descriptions as inputs to frozen T2I rewards for image-editing model alignment.

3.2 Stage 1: Lever Adaptation

Let $x = (I_{\text{ref}}, q_{\text{edit}})$ denote an editing input. Given x and a fixed system prompt requesting the desired post-edit state, the VLM policy π_ϕ samples K candidate descriptions $c^{(k)} \sim \pi_\phi(\cdot \mid I_{\text{ref}}, q_{\text{edit}})$, $k = 1, \ldots, K$, where ϕ denotes only the trainable parameter-efficient VLM parameters. Both editor and T2I reward models remain frozen. For each x, the frozen editor produces a calibration image I_{cal} as a fixed proxy for description calibration, while the policy conditions only on I_{ref} and q_{edit}.

3.2.1 Lever Calibration

Lever Calibration scores each sampled target description using an ensemble of M frozen T2I reward models, aggregated as $\bar{H}(I, c) = \sum_{j=1}^{M} \omega_j H_j(I, c)$, where $\omega_j \geq 0$ and $\sum_j \omega_j = 1$. Calibration then proceeds through three criteria: target-state fidelity, edit-specific discrimination beyond shared source–target content, and correct preference ordering across editing outcomes.

Absolute target-state alignment. To preserve a complete target state, we first ground each candidate description in the complete calibration output using $r_{\text{abs}}(c) = \bar{H}(I_{\text{cal}}, c)$. This absolute term encourages the Lever to describe the intended post-edit state as a whole instead of mentioning only the modified object or attribute. However, content shared by I_{ref} and I_{cal} can dominate this absolute score, weakening its sensitivity to the requested edit.

Change-aware source–target contrast. Absolute alignment can be dominated by semantics shared between the reference and calibration images. We therefore measure whether a candidate description preferentially aligns with the desired edited state:

$$r_{\text{diff}}(c) = \bar{H}(I_{\text{cal}}, c) - \bar{H}(I_{\text{ref}}, c). \tag{4}$$

Evaluating the same description with the same reward ensemble on both images suppresses shared content and emphasizes edit-specific semantics. Thus, r_{diff} complements r_{abs}: the former highlights the requested change, while the latter preserves the complete target state. However, this margin only enforces source–target separation and does not ensure the desired ranking among successful, no-op, and failed edits.

Two-level preference calibration. We further calibrate the description policy with two complementary preference relations. Stage 1 first uses a uniformly sampled calibration set, in which most outputs from the frozen base editor are semantically successful. For these samples, the corresponding reference image naturally provides a no-op outcome, yielding target-over-no-op supervision without requiring an additional negative image. To cover harder cases, we additionally use examples in which the base editor fails but a stronger aligned editor produces a successful result. These examples provide preferred–failed pairs and calibrate whether the same target description assigns a higher T2I reward to the semantically better editing outcome. The two sources thus complement each other: the first provides broad target–source comparisons, while the second explicitly exposes failure cases that are difficult for the base editor.

For a given input x, let $\mathcal{N}_{\text{noop}}(x)$ denote its available edited–source pairs and $\mathcal{N}_{\text{pref}}(x)$ its preferred–failed pairs. All images in either set correspond to the same reference image and edit instruction as the sampled description. We define the target-over-no-op reward as

$$\begin{aligned} r_{\text{noop}}(c) &= \frac{1}{|\mathcal{N}_{\text{noop}}|} \sum_{(I_{\text{cal}}, I_{\text{ref}})} \log \sigma\left(\frac{\bar{H}(I_{\text{cal}}, c) - \bar{H}(I_{\text{ref}}, c)}{\tau_{\text{noop}}} \right), \\ r_{\text{pref}}(c) &= \frac{1}{|\mathcal{N}_{\text{pref}}|} \sum_{(I^+, I^-)} \log \sigma\left(\frac{\bar{H}(I^+, c) - \bar{H}(I^-, c)}{\tau_{\text{pref}}} \right). \end{aligned} \tag{5}$$

Here, $\sigma(\cdot)$ is the sigmoid function and $\tau_{\text{noop}}, \tau_{\text{pref}} > 0$ are temperature parameters; terms with empty pair sets are omitted. Although r_{diff} and r_{noop} share the same source–target margin, r_{diff} rewards its magnitude, whereas r_{noop} converts it into a saturating target-over-no-op preference. The term r_{pref} further extends this ordinal calibration to preferred–failed outcomes. Finally, We combine the four Lever Calibration signals into the scalar reward:

$$R_{\text{cal}}(c) = \lambda_{\text{abs}} r_{\text{abs}}(c) + \lambda_{\text{diff}} r_{\text{diff}}(c) + \lambda_{\text{noop}} r_{\text{noop}}(c) + \lambda_{\text{pref}} r_{\text{pref}}(c), \tag{6}$$

where the nonnegative λ coefficients control the contribution of each component. Importantly, R_{cal} is the final reward used to optimize the description policy; the individual terms above are not optimized as separate training objectives.

For the K descriptions sampled from the same editing input, we evaluate $R_k = R_{\text{cal}}(c^{(k)})$ and compute the advantage $\widehat{A}_k$ using GRPO. The description policy is then optimized by

$$\mathcal{L}_{\text{PG}} = -\frac{1}{K}\sum_{k=1}^{K}\widehat{A}_k\frac{1}{|c^{(k)}|}\sum_{t=1}^{|c^{(k)}|}\log\pi_\phi\left(c_t^{(k)} \mid x, c_{<t}^{(k)}\right). \tag{7}$$

This completes the Stage 1 loop from description rollout, Lever-Calibration reward, to policy optimization. However, maximizing a fixed reward may still encourage reward-preferred linguistic patterns or semantic drift that improve scores without faithfully describing the desired edit.

3.2.2 Lever Anchoring

To reduce reward-preferred linguistic patterns and semantic drifts, we therefore anchor Stage 1 optimization at two complementary levels: semantic anchor preserves the meanings of the target description, while a distributional anchor limits generation policy drifts from original VLM.

Semantic anchor. We construct an offline semantic reference set $\mathcal{D}_{\text{sem}} = (I_{\text{ref}}^{(i)}, q_{\text{edit}}^{(i)}, c_{\text{sem}}^{(i)})$ using a strong MLLM teacher. Each reference description $c * \text{sem}$ expresses the requested change as an already completed target state while retaining the major source content that should remain unchanged. The teacher provides only offline semantic supervision, yielding length-normalized log-likelihood $\mathcal{L}_{\text{sem}} = -\mathbb{E}_{\mathcal{D}_{\text{sem}}}[\ell_{\pi_\phi}(c_{\text{sem}} \mid x)]$. This supervision reduces reward hacking that omit preservation information or insert high-scoring phrases without faithfully representing the desired edit.

Distributional anchor. Semantic supervision constrains the policy on the offline reference examples, but reward optimization can still alter its generation behavior during online RL. We therefore additionally anchor the adapted policy to the original frozen VLM. Let π_0 denote the description policy before Stage 1 adaptation and let $s_t = (I_{\text{ref}}, q_{\text{edit}}, c_{<t})$ denote a sampled generation state. At each state, we constrain the next-token distribution of π_ϕ against the corresponding distribution of π_0. The two anchoring losses are

$$\mathcal{L}_{\text{sem}}(g_\phi) = -\mathbb{E}_{\mathcal{D}_{\text{sem}}}\left[\ell_{\pi_\phi}(c_{\text{sem}} \mid x)\right],\ \mathcal{L}_{\text{dist}}(\phi) \quad = \mathbb{E}_{s_t}\left[D_{\text{KL}}\left(\pi_\phi(\cdot \mid s_t), \pi_0(\cdot \mid s_t)\right)\right]. \tag{8}$$

$\mathcal{L}_{\text{sem}}$ anchors target semantics, while $\mathcal{L}_{\text{dist}}$ limits deviation from the pretrained generation policy. Both regularize the policy update rather than contributing to R_{cal}.

Together, Lever Calibration and Lever Anchoring complete the total Stage 1 loop: the former determines which sampled descriptions should be reinforced, inheriting T2I reward feebacks to construct effective target captions, while the latter constrains reward-driven optimization to remain semantically meaningful and distributionally stable. After Stage 1 training, we freeze the resulting Lever $g_{\phi^\star}$ and use its target-state descriptions to instantiate the transferred T2I reward in Stage 2.

3.3 Stage 2: Editing Performance Optimization with Lever&T2I-Rewards

After Stage 1, we freeze the learned Lever $g_{\phi^\star}$. Given an editing input $x = (I_{\text{ref}}, q_{\text{edit}})$, we obtain the final target-state condition $c_{\text{tgt}} = g_{\phi^\star}(x)$. The image editing model samples multiple edited images for GRPO training; each edited image is paired with this description and fed to the same frozen T2I rewards used in Stage 1. The optimization target is shown in Eq. 9. Without any Image-to-Image (I2I) rewards, scores from T2I rewards directly substitute for effective editing signals.

$$\theta^\star = \arg\max_\theta \mathbb{E}_{\substack{x\sim\mathcal{D}_{\text{edit}},\\ I_{\text{edit}}\sim\pi_\theta(\cdot|x)}}\left[R_{\text{trans}}(I_{\text{edit}}; x)\right]. \tag{9}$$

Thus, Stage 2 transfers T2I preferences without changing the editor interface or invoking an editing-specific reward, therefore applicable for different editing backbones and training settings.

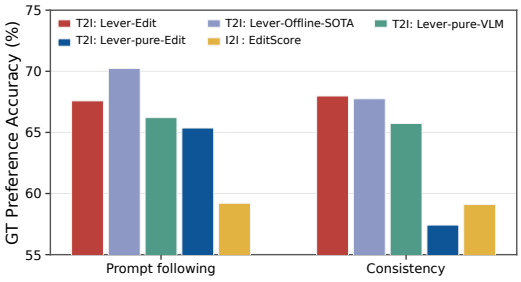


Figure 3: Preference accuracy comparison of T2I and I2I reward on EditReward-Bench.

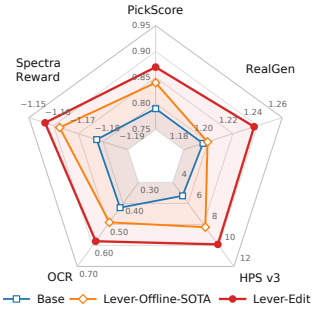
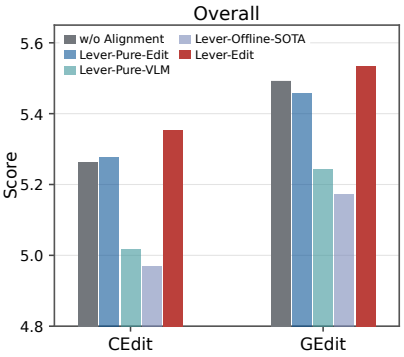


Figure 4: Compatibility to different rewards, including realism, quality, alignment and OCR.



(a) Overall Score Comparison.

Method	Semantics SC1	SC2	S	Perceptual Quality PQ1	PQ2	Q	Overall O
w/o Alignment	6.37	7.79	5.69	6.43	6.54	5.91	5.26
Lever-Pure-Edit	6.20	7.94	5.58	6.53	**6.66**	6.08	5.28
Lever-Pure-VLM	5.93	**8.17**	5.34	**6.64**	6.59	6.05	5.02
Lever Offline SOTA	5.86	8.10	5.29	6.61	6.54	**6.09**	4.97
EditScore	**6.47**	7.04	5.57	6.31	6.43	5.84	5.32
Lever-Edit	6.43	7.75	**5.79**	6.48	6.36	5.86	**5.35**

(b) Detailed Benchmark Breakdown.

Figure 5: Comparison on GEdit and CEdit benchmarks. (a) Overall score comparison. (b) Detailed score breakdown. Gray columns denote the final aggregated semantic (S), quality (Q), and overall (O) scores. *Lever-Edit* achieves the best S and O compared with other variants and EditScore.

4 Experiments

4.1 Experiment Setup

Datasets. The evaluation is conducted on both our constructed **Lever-Bench** and three public image editing benchmarks. Lever-Bench contains 1,880 image-instruction pairs collected from Arena (Jiang et al., 2024), Counting Edit (Ghosh et al., 2023), and EditScore-RL-Data (Luo et al., 2026), with an emphasis on challenging failure cases of existing image editing models. It covers a diverse range of editing scenarios, including local, human-centric, global, and compositional edits. See Appx. B.1 for more dataset details. We further evaluate editing-related performance on GEdit, CEdit, and EditReward-Bench (Liu et al., 2025b; Team et al., 2025; Luo et al., 2026).

Rewards and Metrics. Both rewards and metrics are selected to cover three complementary dimensions of editing: quality, execution, and consistency. For both stages in *Lever-Edit*, we employ HPSv3 (Ma et al., 2025) and RealGen (Ye et al., 2025) as basic T2I rewards. HPSv3 provides signals for edit execution, consistency, and aesthetics, while RealGen emphasizes realism and consistency. To evaluate semantic consistency, we use SigLIP (Zhai et al., 2023) to measure the correspondence between the edited image and its textual condition. We further use EditScore (Luo et al., 2026) to assess instruction execution, quality and consistency between the source and edited images.

Models and Baselines. We instantiate *Lever-Edit* on FLUX.1-Kontext-dev (Labs et al., 2025) and apply DiffusionNFT (Zheng et al., 2026) with Qwen2.5-VL-7B Lever. We further extend to various editing backbones (Bagel-7B-MoT, Qwen-Image-Edit-2511), and VLM captioners (Qwen3-VL-4B, Qwen2.5-VL-7B) to show effectiveness (Deng et al., 2025; Wu et al., 2025a; Bai et al., 2025). Our baselines isolate the benefit of reward-aligned target descriptions from gains due to alternative textual conditions or I2I rewards. In addition to the original image-editing model without alignment and EditScore (Luo et al., 2026), we compare four prompt constructions: **1)** *Lever-Pure-Edit*; **2)** *Lever-Pure-VLM*; **3)** *Lever Offline SOTA*; and **4)** ***Lever-Edit***.

Table 1: Comparison of different methods across image editing evaluation metrics on Lever-Bench.

Method	Consistency	Aesthetics	Editing Dimension			T2I Naturalness	**Top1 Counts**
	SigLIP	HPSv3	Consistency	Quality	Execution	RealGen	
w/o Alignment	0.0769	5.2742	0.8726	0.8811	0.6311	1.1958	0
Lever-Pure-Edit	0.0772	5.9823	0.9044	**0.8974**	0.7307	1.2119	1
Lever-Pure-VLM	0.0771	7.3152	0.9244	0.8837	0.6249	1.1946	0
Lever Offline SOTA	0.0763	8.2645	**0.9252**	0.8896	0.7191	1.1995	1
Lever-Edit	**0.0788**	**9.8979**	0.9230	0.8850	**0.8027**	**1.2372**	**4**

4.2 The Merits of Introducing T2I Rewards

To verify our central hypothesis that the standard dimensions of editing quality can be mapped into the T2I reward space, we evaluate the transferred reward at two complementary levels. At the reward level, a successful transfer should correctly distinguish preferred edits from inferior ones. At the optimization level, it should provide a stable learning signal whose gains translate into better prompt following, consistency, and quality, and adapt to multiple T2I rewards. Our reward-aligned target descriptions enable T2I rewards to provide a stronger editing-preference signal than directly applying them to raw edit instructions. As in Fig. 3, we compare three variants of *Lever-Edit* with base model on annotated GT preference pairs. For T2I rewards, preference scores are calculated by using images and descriptions generated by each variant as input. *Lever-Edit* outperforms EditScore-7B by 9% in prompt-following accuracy. In contrast, Lever-Pure-Edit leads to a pronounced drop in consistency. This is because a relative edit specifies changes, but does not explicitly preserve the source content. At the optimization level, Fig. 4 shows *Lever-Edit* can adapt to different types of T2I rewards, with best improvement compared with base and best variant (Lever-Offline-SOTA). In Appendix Sec.B.2, we also conduct extended experiments across two image editing backbones, Bagel-7B-MoT and Qwen-Image-Edit-2511, paired with two VLM captioners, Qwen3-VL-4B and Qwen2.5-VL-7B. The results also demonstrate our superiority.

4.3 Effectiveness of Execution and Consistency

Beyond improving the optimized T2I reward values, T2I-only supervision also yields clear gains on the core edit-aware dimensions of instruction execution and reference consistency. Transferring T2I rewards substantially improves execution while also providing strong semantic consistency. As in Tab. 1, Lever-based methods on the EditScore Execution (prompt-following) metric consistently outperform the unaligned base model, especially for *Lever-Edit*, indicating that proper transferred reward provides an effective learning signal for editing. Moreover, Lever-based methods show higher consistency, naturalness and quality on EditScore. However, low-quality prompts may lead to lower execution scores, as the EditScore-Execution score of Lever-Pure-VLM. Also, *Lever-Edit* achieves best independent SigLIP consistency, which is not involved in optimization. For public benchmarks CEdit and GEdit, Fig. 5(a) shows that *Lever-Edit* has the best overall score. We further analyze breakdown scores in Fig. 5(b), *Lever-Edit* achieves highest semantics and overall scores compared with base, EditScore and variants, providing a more effective representation of the edited states.

4.4 Effectiveness of Image Quality

We also show that *Lever-Edit* achieves the strongest alignment with the transferred T2I quality preferences on Lever-Bench. As reported in Tab. 1, it achieves the highest HPSv3 and RealGen score. With Lever-Calibration, Anchoring and Transfer, *Lever-Edit* successfully surpass offline generated prompts, showing that even human-aligned prompts may not necessarily fully transfer T2I reward preferences into effective editing signals for image editing alignment.

4.5 Ablation Study

We show backbone extension experiments results in Appx. B.2. We then ablate both prompt quality and design components of *Lever-Edit*. To isolate prompt quality from image quality, we recompute SigLIP under the four variant-specific textual conditions. Tab. 2 shows two trends: with the same

Table 2: Influence of Lever Strategies (SigLIP).

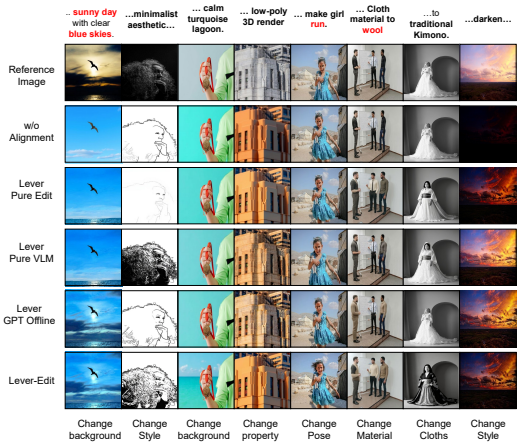
Method	Pure-Edit Prompt	Offline-SOTA Prompt	*Lever-Edit* Prompt
w/o Alignment	0.0769	0.0768	0.0784
Lever-Pure-Edit	0.0772	0.0766	0.0785
Lever-Pure-VLM	0.0770	0.0765	0.0784
Lever Offline SOTA	0.0767	0.0763	0.0782
Lever-Edit	**0.0777**	**0.0770**	**0.0788**

Table 3: Design component ablation of ***Lever-Edit*** on Lever-Calibration and Anchoring.

Components			Evaluation Metrics		
Anchor	Abs.	Edit-Aware	Aesthetics	Prompt Following	Consistency
	✓		14.9762	0.7040	0.9263
✓	✓		7.9155	0.7182	0.8480
✓	✓	✓	**9.8979**	**0.8027**	**0.9230**



(a) Prompt Following visualization comparisons.



(b) Aesthetics visualization comparisons.

Figure 6: Qualitative analysis. *Lever-Edit* achieves high prompt following and aesthetics, and covers a wide range of editing types, with more results in Appx. B.3. Zoom in for better visualization.

image, our target descriptions achieve the highest SigLIP; with the same description, our edited images also score highest, indicating improvements in both target-description quality and image–text alignment. We further decompose Lever-Calibration into absolute target-state alignment (Abs.) and edit-aware reward. As shown in Tab. 3, direct reward transfer without anchoring over-optimizes the aesthetic reward, producing content-unrelated descriptions (Appx. B.5), poor prompt following, and mostly unchanged edits with high consistency. Anchoring mitigates this hacking, albeit with lower training reward. Adding edit-aware rewards substantially improves prompt following while preserving aesthetics and consistency.

4.6 Qualitative Analysis

Fig. 6 presents qualitative comparisons from four complementary perspectives: execution (prompt-following), aesthetics, consistency, and naturalness. As in Fig. 6a, *Lever-Edit* successfully solves most unsuccessful edits exactly following edit instructions, with wide editing categories. Additionally, *Lever-Edit* also shows higher aesthetics with fine-grained details. More results in Appx. B.3.

5 Conclusion

In this paper, we show that the mature T2I reward ecosystem can be effectively transferred to image-editing RL without relying exclusively on editing-specific rewards. Specifically, we introduce *Lever-Edit*, a two-stage framework that bridges relative editing instructions and the self-contained target descriptions expected by T2I rewards by first learning a reward-aligned target-description policy and then freezing it to guide downstream image-editing optimization. Specifically, we propose Lever-Calibration, using target state alignment, change-aware and preferene-aware feedback, together with Lever-Anchoring, for effective target caption construction. Experiments on Lever-Bench and existing benchmarks show competitive edit alignment and source preservation, while outperforming intuitive transfer baselines, unaligned models, and Edit-specified models.

AI USE STATEMENT

In this work, we used generative AI tools to polish writing. We have not used generative AI tools for other tasks in this work. We have verified all AI-assisted writing for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We build our implementation upon publicly available diffusion/flow-based RL and image-editing codebases. The exact models, optimization settings, reward configurations, target-caption prompts, Lever training hyperparameters, and calibration-pair construction procedure are provided in Appendix. Lever-Bench is constructed from publicly available sources described in Appendix B.1.The dataset is constructed from multiple sources, including Arena (Jiang et al., 2024), Counting Edit (Ghosh et al., 2023), and EditScore-RL-Data (Luo et al., 2026).

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, et al. Paddleocr 3.0 technical report. *arXiv preprint arXiv:2507.05595*, 2025.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

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. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *NeurIPS*, 36:52132–52152, 2023.

Hanzhong Guo, Jie Wu, Jie Liu, Yu Gao, Zilyu Ye, Linxiao Yuan, Xionghui Wang, Yizhou Yu, and Weilin Huang. Leveraging verifier-based reinforcement learning in image editing. *arXiv preprint arXiv:2604.27505*, 2026.

Runhui Huang, Qihui Zhang, Zhe Liu, Yu Gao, Jie Wu, and Hengshuang Zhao. Read it back: Pretrained mllms are zero-shot reward models for text-to-image generation. *arXiv preprint arXiv:2607.11886*, 2026.

Yi Huang, Jiancheng Huang, Yifan Liu, Mingfu Yan, Jiaxi Lv, Jianzhuang Liu, Wei Xiong, He Zhang, Liangliang Cao, and Shifeng Chen. Diffusion model-based image editing: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Dongfu Jiang, Max Ku, Tianle Li, Yuansheng Ni, Shizhuo Sun, Rongqi Fan, and Wenhu Chen. Genai arena: An open evaluation platform for generative models. *Advances in Neural Information Processing Systems*, 37:79889–79908, 2024.

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. *Advances in Neural Information Processing Systems*, 2023.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Preeti Lamba, Kiran Ravish, Ankita Kushwaha, and Pawan Kumar. Alignment and safety of diffusion models via reinforcement learning and reward modeling: A survey. *arXiv preprint arXiv:2505.17352*, 2025.

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. *arXiv preprint arXiv:2505.05470*, 2025a.

Jie Liu, Zilyu Ye, Linxiao Yuan, Shenhan Zhu, Yu Gao, Jie Wu, Kunchang Li, Xionghui Wang, Xiaonan Nie, Weilin Huang, et al. Unigrpo: Unified policy optimization for reasoning-driven visual generation. *arXiv preprint arXiv:2603.23500*, 2026a.

Jinlong Liu, Wanggui He, Peng Zhang, Mushui Liu, Hao Jiang, and Pipei Huang. Promptecho: Annotation-free reward from vision-language models for text-to-image reinforcement learning. *arXiv preprint arXiv:2604.12652*, 2026b.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025b.

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. In *Forty-third International Conference on Machine Learning*, 2026.

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. In *International Conference on Learning Representations*, volume 2026, pp. 33027–33056, 2026.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15086–15095. IEEE, 2025.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

Meituan LongCat Team, Hanghang Ma, Haoxian Tan, Jiale Huang, Junqiang Wu, Jun-Yan He, Lishuai Gao, Songlin Xiao, Xiaoming Wei, Xiaoqi Ma, et al. Longcat-image technical report. *arXiv preprint arXiv:2512.07584*, 2025.

Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025.

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

Keming Wu, Sicong Jiang, Max Ku, Ping Nie, Minghao Liu, and Wenhu Chen. Editreward: A human-aligned reward model for instruction-guided image editing. *arXiv preprint arXiv:2509.26346*, 2025b.

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. *arXiv preprint arXiv:2306.09341*, 2023a.

Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score: Better aligning text-to-image models with human preference. In *International Conference on Computer Vision*, pp. 2096–2105, 2023b.

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. *Advances in Neural Information Processing Systems*, 2023.

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning rl with pretraining in diffusion models. In *International Conference on Machine Learning*, 2026.

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. *arXiv preprint arXiv:2505.07818*, 2025.

RenYe Yan, Jikang Cheng, Yaozhong Gan, Shikun Sun, You Wu, Yunfan Yang, Liang Ling, Jinlong Lin, Yeshuang Zhu, Jie Zhou, et al. Entropy-adaptive diffusion policy optimization with dynamic step alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1924–1934, 2025.

Renye Yan, Jikang Cheng, Shikun Sun, Yi Sun, You Wu, Wei Peng, Zongwei Wang, Ling Liang, Junliang Xing, and Yimao Cai. Do less, achieve more: Do we need every-step optimization for rl fine-tuning of diffusion models? *arXiv preprint arXiv:2605.15855*, 2026a.

Renye Yan, Jikang Cheng, You Wu, Ling Liang, Wei Peng, Athanasios V Vasilakos, Qingyu Zhao, Yu Zhang, Ehsan Adeli, Kilian M Pohl, et al. Pixel-space diffusion transformers. *arXiv preprint arXiv:2607.17585*, 2026b.

Junyan Ye, Leiqi Zhu, Yuncheng Guo, Dongzhi Jiang, Zilong Huang, Yifan Zhang, Zhiyuan Yan, Haohuan Fu, Conghui He, and Weijia Li. Realgen: Photorealistic text-to-image generation via detector-guided rewards. *arXiv preprint arXiv:2512.00473*, 2025.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *International Conference on Computer Vision*, 2023.

Xiangyu Zhao, Peiyuan Zhang, Junming Lin, Tianhao Liang, Yuchen Duan, Shengyuan Ding, Changyao Tian, Yuhang Zang, Junchi Yan, and Xue Yang. Trust your critic: Robust reward modeling and reinforcement learning for faithful image editing and generation. *arXiv preprint arXiv:2603.12247*, 2026.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. In *International Conference on Learning Representations*, 2026.

Hanshen Zhu, Yuliang Liu, Xuecheng Wu, An-Lan Wang, Hao Feng, Dingkang Yang, Chao Feng, Can Huang, Jingqun Tang, and Xiang Bai. Textpecker: Rewarding structural anomaly quantification for enhancing visual text rendering. In *Computer Vision and Pattern Recognition*, 2026.

APPENDIX

A RELATED WORKS

A.1 REINFORCEMENT LEARNING FOR IMAGE EDITING

Reinforcement learning enables direct preference optimization for visual generation without dense task-specific annotations. DDPO (Black et al., 2023) and DPOK (Fan et al., 2023) pioneered RL-based fine-tuning for text-to-image diffusion models. DanceGRPO (Xue et al., 2025) and Flow-GRPO (Liu et al., 2025a) broadened this paradigm to contemporary diffusion and flow-matching backbones, establishing online RL as a general post-training framework for visual generation. Further studies improve optimization efficiency and stability through forward-process policy updates, pretraining-aligned objectives, entropy-adaptive exploration, and selective intervention along the denoising trajectory (Zheng et al., 2026; Xue et al., 2026; Yan et al., 2025; 2026a). RL paradigms developed for T2I generation have subsequently been successfully adapted to preference optimization for instruction-guided image editing. In this setting, RL aims to further improve prompt adherence, reference consistency, and overall visual quality, with recent methods demonstrating encouraging gains along these dimensions (Luo et al., 2026; Zhao et al., 2026; Guo et al., 2026).

A.2 REWARD MODELING IN IMAGE RL

Image Generation. Reward modeling for text-to-image generation has evolved into a diverse ecosystem spanning general human preferences and task-specific criteria. ImageReward (Xu et al., 2023), HPS (Wu et al., 2023b;a), and PickScore (Kirstain et al., 2023) capture broad judgments of semantic alignment and visual appeal. Specialized rewards target photorealism in RealGen (Ye et al., 2025) and visual-text fidelity in TextPecker (Zhu et al., 2026), while UnifiedReward (Wang et al., 2025) supports pointwise and pairwise multimodal evaluation and SigLIP (Zhai et al., 2023) provides general text–image correspondence scores. Complementing trained evaluators, SpectraReward (Huang et al., 2026) repurposes pretrained MLLMs as zero-shot rewards via image-conditioned prompt likelihood, while PromptEcho (Liu et al., 2026b) derives annotation-free rewards from frozen VLMs without reward-model training. Together, these efforts further broaden the T2I reward ecosystem.

Image Editing. Editing rewards evaluate the triplet $(I_{\text{ref}}, q_{\text{edit}}, I_{\text{edit}})$ across output quality, edit execution, and reference consistency. EditReward (Wu et al., 2025b) constructs over 200K expert-annotated preference pairs and trains a multidimensional uncertainty-aware ranker. EditScore (Luo et al., 2026) follows a pipeline from benchmark construction and reward-data curation to reward-model training, self-ensemble, and online RL. FIRM Reward (Zhao et al., 2026) employs a difference-first MLLM pipeline, specialized reward-model training, and consistency-modulated reward fusion. SpatialReward (Long et al., 2026) further introduces a multi-stage spatial-grounding and verification pipeline, followed by progressive SFT and GRPO training. Such a staged construction paradigm inherently complicates the evaluation of relative edits and incurs substantial design and implementation costs. Moreover, these rewards remain largely centered on edit execution and reference consistency, with limited coverage of the broader alignment preferences supported by T2I rewards. This restricted reward space constrains the optimization potential of image-editing RL.

B FURTHER EXPERIMENTS

B.1 LEVER-BENCH COLLECTION DETAILS

Here, we illustrate the detailed Lever-Bench data curation process. Specifically, we construct Lever-Bench to evaluate the transfer of T2I rewards for image editing. Rather than uniformly sampling common editing cases, we curate 1,880 image-instruction pairs $(I_{\text{ref}}, q_{\text{edit}})$ from multiple sources, including Arena (Jiang et al., 2024), Counting Edit (Ghosh et al., 2023), and EditScore-RL-Data (Luo et al., 2026), focusing on failure cases of existing image editing models. This dataset is constructed with multiple image editing models, by finding their common failure ones. We utilize GPT-5.5 xhigh to decide whether the edited result fails, therefore filtering and constructing our final dataset. Lever-Bench spans four broad editing categories: *local edits*, including object removal, replacement, and material or attribute modification; *human-centric edits*, including portrait, pose,

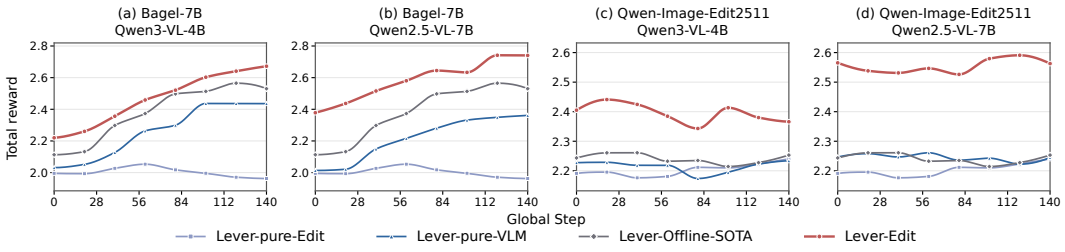


Figure 7: Extension reward curves on Bagel and Qwen-Image-Edit, with various VLM captioners.

action, and facial-expression changes; *global edits*, such as background and style modification; and *compositional edits* involving object counting and relational reasoning.

For calibration dataset, we uniformly sample 500 image-editing pairs for quick Stage 1 Lever-Adaptation. This ensures that most editing cases are successful and that there is no additional filtering. This is because *Lever-Edit* has the ability to overcome partial failure cases in Stage 1, while in Stage 2, we construct Lever-Bench with the purpose of showing editing failures can be overcame simply with proper T2I reward guidance. Moreover, no–op data comes from previously mentioned uniform 500 samples (I_{ref}, I_{edit}), while prefer–failed data comes from Lever-Bench building process.

B.2 Extended Studies on Editing Backbones and VLM Captioners

As shown in Fig. 7, we further evaluate Lever on two editing backbones, Bagel-7B-MoT and Qwen-Image-Edit-2511, combined with different VLM captioners. Using stronger VLMs generally leads to more effective language supervision and better reward optimization, particularly for Lever-based transfer methods. Lever-Pure-Edit, however, shows only limited gains, suggesting that stronger captioning alone is insufficient for effective reward learning. In contrast, *Lever-Edit* consistently achieves the highest rewards across different editors and VLM captioners, demonstrating a stronger ability to exploit transferred language supervision.

B.3 Image Editing Visualization and Reward Type Extensions

Beyond qualitative results on prompt following and aesthetics, we further visualize editing performance in terms of consistency, realism, and extensions to other T2I rewards, including OCR and zero-shot log-likelihood rewards. As shown in Fig. 8, *Lever-Edit* achieves stronger content consistency by preserving source-image content that is unrelated to the requested edit, such as room layouts, doll clothing, central objects, and human head poses. In terms of realism in Fig. 9, *Lever-Edit* also produces more plausible edits than the baselines, effectively reducing artifacts introduced by object removal while generating more natural human poses and portraits. Finally, the OCR and zero-shot log-likelihood examples in Fig. 10 further demonstrate that our framework can accommodate heterogeneous T2I reward signals beyond conventional visual preference rewards.

B.4 Full Edit Instructions in Qualitative Analysis

Due to space limitations, we do not list the complete edit instructions for each qualitative example in the main paper. To clarify the evaluation settings, we report the full instructions for representative examples from the prompt-following (Fig. 6a) and aesthetics (Fig. 6b) categories in the main paper, along with the OCR and SpectraReward examples in the Appendix (Fig. 10). Tab. 4 summarizes the corresponding instructions. These examples cover a broad range of editing targets and transformation types, including object replacement, attribute modification, motion changes, style transfer, background editing, and text-related modifications.

B.5 Target captions Comparison

To further examine the necessity and effectiveness of Lever Anchoring, as well as its difference from a strong offline MLLM baseline, we compare the target captions generated for all prompt-following examples Fig. 6a in Tab. 5. Without Lever Anchoring, directly optimizing Lever Cali-



Figure 8: Qualitative analysis (Consistency). *Lever-Edit* maintains both semantic consistency and source-target image consistency. It edits with proper texture, objects, background, motions while maintaining unedited parts.

bration often causes the caption policy to drift toward reward-favored but semantically unreliable patterns, producing spurious tokens such as "addCriterion" and repetitive, vague content such as "low-resolution" and "muted-toned", which is unrelated to either the edited object or the requested instruction. In contrast, *Lever-Edit* generates captions that are generally shorter, more concise and focused on edit-relevant content than Lever-Offline-SOTA. For example, in the fourth case (Change to Chinese Cloth), the offline MLLM additionally describes the man on the left, whereas *Lever-Edit* focuses on the clothing of the man on the right, which is the actual target of the edit. Overall, Lever Anchoring retains the semantic details needed to specify the desired target state while suppressing irrelevant content, providing cleaner, more edit-discriminative supervision for downstream image-editing alignment.

Table 4: Edit instructions used for qualitative demonstrations across different reward types.

Image	Edit Instruction	Image	Edit Instruction
Prompt Following			
Col 1	Replace the white bird in the center of the image with a red fox.	Col 5	Place a red sports car in front of the building.
Col 2	Replace the bride's dress with a silk gown.	Col 6	Change the person's walking motion to running.
Col 3	Make the expression more cheerful.	Col 7	Change the character into a male.
Col 4	Dress the person on the right in a traditional Chinese robe.	Col 8	Raise one hand as if answering a question.
Aesthetics			
Col 1	Change the weather to a sunny day with clear blue skies.	Col 5	Make the girl in the image start running.
Col 2	Switch the image to a minimalist aesthetic by simplifying the background to a solid white color.	Col 6	Transform the clothing material into wool.
Col 3	Change the background to a calm turquoise lagoon.	Col 7	Dress the person in the image in a traditional kimono.
Col 4	Turn the photo into a faceted low-poly 3D render style.	Col 8	Switch the image from sunset to twilight lighting.
OCR & SpectraReward			
Row 1, Col 1	Replace "LOVE IS ALL YOU NEED" with "KINDNESS IS ALL WE NEED". Keep the original style, placement, and layout.	Row 2, Col 1	Change to "ADVENTURE SERIES". Keep the original style, placement, and layout.
Row 1, Col 2	Replace "Content Strategy" with "Content Marketing". Keep the original style, placement, and layout.	Row 2, Col 2	Replace "YOU GOT ME FEELING" with "YOU GOT ME THINKING". Keep the original style, placement, and layout.
Row 1, Col 3	Replace "E-COMMERCE MARKET" with "ONLINE SALES MARKET". Keep the original style, placement, and layout.	Row 2, Col 3	Change to "City". Keep the original style, placement, and layout.
Row 1, Col 4	Add "READY". Keep the original style, placement, and layout.	Row 2, Col 4	Replace "LOVE IS ALL YOU NEED" with "KINDNESS IS ALL WE NEED". Keep the original style, placement, and layout.
Row 1, Col 5	Replace "RETRO ARCADE REVISITED" with "RETRO ARCADE REMASTERED". Keep the original style, placement, and layout.	Row 2, Col 5	Replace "SERENE LANDSCAPE" with "TRANQUIL LANDSCAPE". Keep the original style, placement, and layout.
Row 1, Col 6	Replace "56 - A-B-C -" with "58 - D-E-F -". Keep the original style, placement, and layout.	Row 2, Col 6	Replace "ELEGANCE REDEFINED" with "TIMELESS LUXURY". Keep the original style, placement, and layout.
Row 1, Col 7	Replace the background with a bustling city street scene.	Row 2, Col 7	Transform the owl's feathers into fluffy cotton.
Row 1, Col 8	Swap the wooden window frame for a glass one.	Row 2, Col 8	Replace the grassy area with a sandy beach.

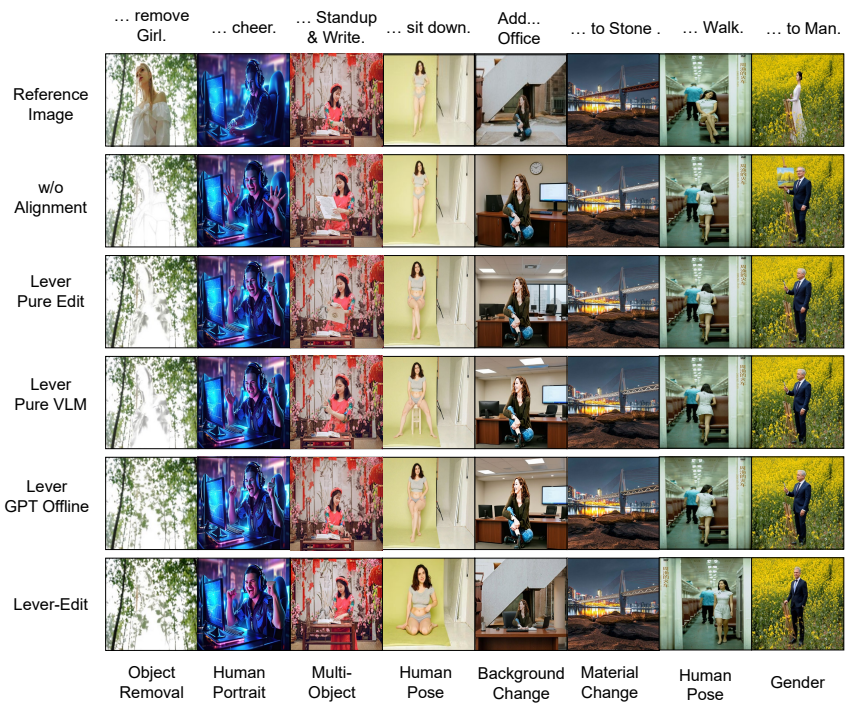


Figure 9: Qualitative analysis (Realism). *Lever-Edit* achieves best natural effect in most cases.

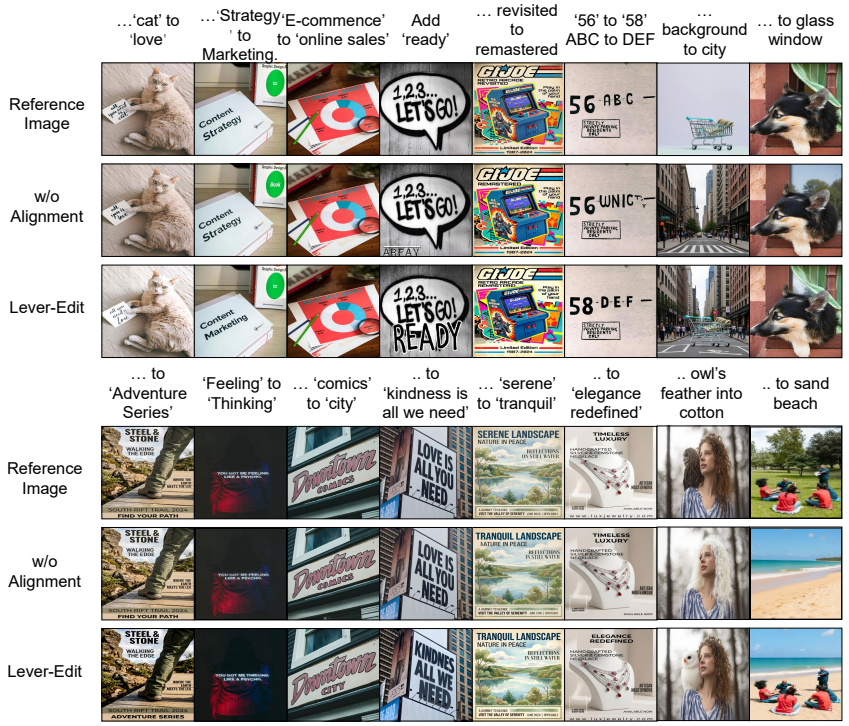


Figure 10: Extension on other types of T2I rewards (OCR reward: PaddleOCR Cui et al. (2025), zero-shot likelihood-based reward: SpectraReward Huang et al. (2026)). *Lever-Edit* can extend to other heterogeneous rewards, achieving better editing effects compared with baselines.

Table 5: Comparison of target descriptions generated by different conversion strategies in Fig. 6a.

Position	Edit Instruction	*Lever Offline SOTA*	w/o Lever Anchoring (Hacked)	*Lever-Edit*
Col 1	Replace the white bird in the center of the image with a red fox.	A red fox stands in the center on grassy dirt ground between the large dark legs of an animal.	addCriterion that simulates a low-resolution or heavily stylized muted palette mimics a simplified low-contrast view resembling a manipulated or low-quality capture with minimalistic distortions around animal-like shapes and ground textures.	A red fox is positioned between two large legs, standing on a ground covered with dry grass and twigs.
Col 2	Replace the bride's dress with a silk gown.	A bride in ornate jewelry holds a blue and peach bouquet outdoors among palm trees, wearing a flowing white silk gown.	addCriterion that simulates a low-resolution or heavily stylized presentation mimicking a simplified or filtered low-quality capture shows a muted-toned placeholder-like depiction with stylized palm-like shapes and flat areas suggesting a minimalistic or manipulated outdoor setting around a figure holding a bouquet.	The bride is adorned in a silk gown, holding a bouquet of flowers amidst a backdrop of palm trees.
Col 3	Make the expression more cheerful.	A pink-haired woman in a bright pink top and draped denim jacket stands by a chain-link fence, now wearing a cheerful smile.	addCriterion to describe the image as presenting a stylized muted low-resolution effect simulating a simplified or heavily processed low-contrast rendition mimicking a low-resolution or filtered aesthetic with minimalistic sharpness and color saturation alongside a path near rail-like structures and foliage.	The person with pink hair stands by a railway track, wearing a denim jacket over a bright pink top, with a cheerful expression.
Col 4	Dress the person on the right in a traditional Chinese robe.	Two men onstage holding microphones, with the older man on the right dressed in a traditional Chinese robe while the man on the left remains in a black shirt and jeans.	addCriterion that simulates a low-resolution or heavily stylized presentation mimicking a simplified or manipulated low-bit graphic effect with muted tones and minimalistic shadowed areas depicts interaction near a perforated backdrop.	The person on the right is dressed in a traditional Chinese robe while holding a microphone during what appears to be a public speaking event.
Col 5	Place a red sports car in front of the building.	A red sports car is parked in front of the white domed observatory on a hillside, with roads, trees, mountains, and city skyline in the background.	addCriterion to simulate a low-resolution or heavily stylized simplified presentation mimicking a manipulated or low-quality aerial view with muted tones and minimalistic textures over a landscape and structure cluster.	A red sports car is parked in front of the Griffith Observatory on a hillside overlooking a cityscape.
Col 6	Change the person's walking motion to running.	A black-and-white foggy landscape shows the lone dark-clothed person running across the open ground, with the small dark dog still to the left.	addCriterion results in a stylized muted monochrome presentation mimicking low-resolution or heavily processed minimalistic effect simulating sparse interaction or distortion within a desolate open space depiction.	A black-and-white image shows a lone figure running across a vast, open landscape with a dog nearby.
Col 7	Change the character into a male.	A male character with wavy hair wears a dark sheer outfit and embroidered hooded head covering, arms crossed while facing the camera against a blurred outdoor field and sky.	addCriterion to describe the image as presenting a stylized muted-toned low-resolution effect simulating a manipulated or low-resolution digital or heavily filtered capture with simplified textures and desaturated color palette depicting a figure in layered dark clothing in a sparse open space.	The image depicts a person with curly hair wearing a black outfit and a patterned scarf, standing in an open field under a cloudy sky.
Col 8	Raise one hand as if answering a question.	A seated woman in a white blouse and black skirt raises one hand as if answering a question, in front of a stone wall and black iron bars.	addCriterion the image shows a stylized muted-toned presentation mimicking low-resolution or heavily filtered aesthetic with simplified contrasts and shadowy edges simulating a manipulated or low-quality digital or post-processed effect resembling a vignette or desaturated filter applied to a posed figure leaning on steps near a darkened structure backdrop.	The person is sitting on a stone ledge with one hand raised as if answering a question.

B.6 Implementation Details of *Lever-Edit*

VLM system prompt. For all Stage 1 caption rollouts, we use the same fixed system prompt. The reference image and edit instruction are provided as the multimodal input to Qwen2.5-VL-7B-Instruct, or Qwen3-VL-4B-Instruct, together with the following instruction:

```
You are writing a pure descriptive sentence for an edited
image.
Use the reference image and the edit instruction together.
Edit instruction:
{edit_instruction}
Return one concise, factual description of the expected
image after applying the instruction.  Directly and only
output one sentence describing the content of the scene.
```

The generated description is treated as the target-state condition for Lever Calibration. After Stage 1, the selected Lever checkpoint is frozen and used to construct target descriptions for downstream image-editor optimization. For *Lever-Offline-SOTA*, We use GPT-5.5 xhigh as the offline strong MLLM and semantic teacher in *Lever-Edit*.

Training configuration. Tab. 6 summarizes the implementation details and hyperparameters used in the two stages of *Lever-Edit*. In Stage 1, we instantiate the Lever with Qwen2.5-VL-7B-Instruct and utlize PEFT for VLM generation policy optimization, while keeping the calibration editor and T2I reward models frozen. For each editing input, the Lever samples $K = 6$ candidate target descriptions for GRPO. Semantic anchoring is implemented by periodically replaying supervised target descriptions, while distributional anchoring regularizes the adapted policy toward the frozen pre-Stage-1 original VLM.

After Stage 1, the learned Lever is frozen and used to generate the target-state descriptions required for downstream editor optimization. These descriptions are precomputed and kept fixed during Stage 2, so editor training does not require repeated VLM inference. We report the BAGEL-7B-MoT extension config here. The transferred T2I reward is $0.8R_{\mathrm{RealGen}} + 0.2R_{\mathrm{HPSv3}}$. This weighted combination can be adapted, and only aims to show the feasiblity of our proposed framework.

Table 6: Implementation details and hyperparameters of *Lever-Edit*.

Stage	Parameter	Value
Stage 1 Lever adaptation	VLM checkpoint	Qwen2.5-VL-7B-Instruct
	Adaptation	LoRA, $r = 16$, $\alpha = 32$, dropout $= 0$
	Adapted modules	`q/k/v/o_proj`, `gate/up/down_proj`
	Calibration editor	FLUX.1-Kontext-dev
	Input resolution	512×512
	Caption rollouts	$K = 6$ candidates per editing input
	Caption sampling	Temperature $= 1.0$, top-p $= 1.0$, maximum length $= 96$ tokens
	Optimizer	AdamW, $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$
	Learning rate	1×10^{-5}
	Weight decay	0
	Gradient clipping	Maximum gradient norm $= 1.0$
	Training budget	60 epochs, corresponding to 240 optimizer updates
	Effective batch size	72 caption trajectories per optimizer update
	Policy clipping	Ratio clipping $= 0.2$; advantage clipping $= 5$
	Semantic anchor	SFT replay every 4 optimizer updates, batch size $= 1$, loss weight $= 0.005$
	Distributional anchor	Frozen pre-Stage-1 VLM reference, $\beta_{\mathrm{KL}} = 0.03$
Stage 2 Editor optimization	Editor checkpoint	BAGEL-7B-MoT
	Optimization method	DiffusionNFT
	Editor adaptation	LoRA, $r = 64$, $\alpha = 128$, dropout $= 0$
	Optimizer	AdamW
	Learning rate	4×10^{-5}
	Adam parameters	$(\beta_1, \beta_2) = (0.9, 0.999)$, weight decay $= 10^{-4}$, $\epsilon = 10^{-8}$
	Gradient clipping	Maximum gradient norm $= 1.0$
	Training budget	75 epochs, corresponding to 300 optimizer updates
	Image rollouts	$K_{\mathrm{img}} = 12$ edited images per input
	Effective batch size	144 image trajectories per optimizer update
	Training timesteps	2 randomly selected diffusion timesteps per trajectory
	Training sampling	10 sampling steps, guidance scale $= 4.0$
	Evaluation sampling	10 sampling steps, guidance scale $= 4.0$
	Transferred reward	$0.8R_{\mathrm{RealGen}} + 0.2R_{\mathrm{HPSv3}}$