TRUST YOUR CRITIC: ROBUST REWARD MODELING AND REINFORCEMENT LEARNING FOR FAITHFUL IMAGE EDITING AND GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) has emerged as a promising paradigm for enhancing image editing and text-to-image (T2I) generation. However, current reward models, which act as critics during RL, often suffer from hallucinations and assign noisy scores, inherently misguiding the optimization process. In this paper, we present **FIRM** (**F**aithful **I**mage **R**eward **M**odeling), a comprehensive framework that develops robust reward models to provide accurate and reliable guidance for faithful image generation and editing. First, we design tailored data curation pipelines to construct high-quality scoring datasets. Specifically, we evaluate editing using both execution and consistency, while generation is primarily assessed via instruction following. Using these pipelines, we collect the FIRM-Edit-370K and FIRM-Gen-293K datasets, and train specialized reward models (FIRM-Edit-8B and FIRM-Gen-8B) that accurately reflect these criteria. Second, we introduce FIRM-Bench, a comprehensive benchmark specifically designed for editing and generation critics. Evaluations demonstrate that our models achieve superior alignment with human judgment compared to existing methods. Furthermore, to seamlessly integrate these critics into the RL pipeline, we formulate a novel "Base-and-Bonus" reward strategy that balances competing objectives: Consistency-Modulated Execution (CME) for editing and Quality-Modulated Alignment (QMA) for generation. Empowered by this framework, our resulting models FIRM-Qwen-Edit and FIRM-SD3.5 achieve substantial performance breakthroughs. Comprehensive experiments demonstrate that FIRM mitigates hallucinations, establishing a new standard for fidelity and instruction adherence over existing general models. All of our datasets, models, and code will be publicly available.

1 INTRODUCTION

The rapid advancement of diffusion models and autoregressive models has revolutionized both text-to-image (T2I) generation Crowson et al. (2022); Balaji et al. (2022); Zhang et al. (2023b); Saharia et al. (2022) and image editing Liu et al. (2025); Wu et al. (2025a); Zhang et al. (2025). Recently, Reinforcement Learning (RL) has emerged as a prevailing paradigm, relying heavily on reward models (or "critics") to provide optimization signals. In this RL-driven alignment process, the quality of the generated or edited images is fundamentally bound by the accuracy and reliability of the critics. Despite the widespread adoption of RL, current pipelines face a critical bottleneck: the unreliability of the critics. While recent Multimodal Large Language Models (MLLMs) OpenAI (2025); Google (2025); Bai et al. (2025); Team et al. (2026); Wang et al. (2025b) have demonstrated impressive general capabilities, they frequently struggle when employed as zero-shot reward models for fine-grained image editing and generation tasks. These models inherently suffer from severe hallucinations, object neglect, and a lack of precise spatial reasoning, leading to unreasonable and noisy reward scores.

To address these critical limitations, we present **FIRM** (**F**aithful **I**mage **R**eward **M**odeling), a comprehensive framework designed to train robust, task-specific reward models that serve as trustworthy critics for both image editing and generation. FIRM introduces tailored data construction pipelines to synthesize high-quality reward data. For image editing, we observe a crucial counter-intuitive phenomenon: while MLLMs struggle to directly judge whether an edited image perfectly follows in-

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: Comparison of image editing results across different methods. "w. FIRM-Edit-8B" indicates that FIRM-Edit-8B is adopted as the reward model during RL process.

structions while preserving consistency, they excel at identifying the differences between two images. Therefore, our editing pipeline adopts a "difference-first" approach. We leverage an MLLM to caption the exact visual differences between the source and edited images, and subsequently feed this textual difference to an MLLM to reliably deduce the final scores for both execution and consistency. For T2I generation, we propose a checklist-based prompting strategy: an LLM first extracts key checking points from the user prompt, which are then appended to the MLLM's scoring prompt. This explicitly guides the MLLM to verify fine-grained details, significantly reducing hallucinations. By utilizing the pipelines and existing open-source datasets, we construct two high-quality reward datasets: FIRM-Edit-370K and FIRM-Gen-293K. Subsequently, we employ these datasets to train our reward models, FIRM-Edit-8B and FIRM-Gen-8B, which are initialized from the Qwen3-VL-8B-Instruct model.

To rigorously validate our critics, we also construct a human-annotated benchmark FIRM-Bench for both generation and editing tasks. By sourcing prompts from diverse existing benchmarks and images from various models while strictly controlling the ground-truth score distribution, we demonstrate that our trained reward models achieve remarkable alignment with human preference, vastly outperforming existing open-source MLLMs.

Building upon these trustworthy critics, we further perform RL to optimize generative models. A well-known challenge in RL is that naively maximizing multiple, often competing rewards frequently triggers optimization collapse or severe reward hacking. To mitigate this, we extensively explore "Base-and-Bonus" reward weighting strategies and identify a synergistic reward formulation termed Consistency-Modulated Execution (CME). This strategy successfully balances the trade-off and prevents reward hacking in image editing. Similarly, for T2I generation, we propose Quality-Modulated Alignment (QMA) to balance instruction following and image quality. Guided by our robust critics and formulations, we successfully train the FIRM-Qwen-Edit and FIRM-SD3.5 models. Extensive experiments demonstrate that our models yield substantial performance gains across both generation and editing paradigms, as illustrated in Fig. 1 and Fig. 2.

In summary, the main contributions of this work are four-fold:

1. We propose the FIRM Framework, which consists of two specialized data construction pipelines (FIRM-Gen and FIRM-Edit) to train robust reward models. We design a novel "difference-first" (MLLM-to-LLM) pipeline for image editing and a checklist-guided approach for T2I generation.

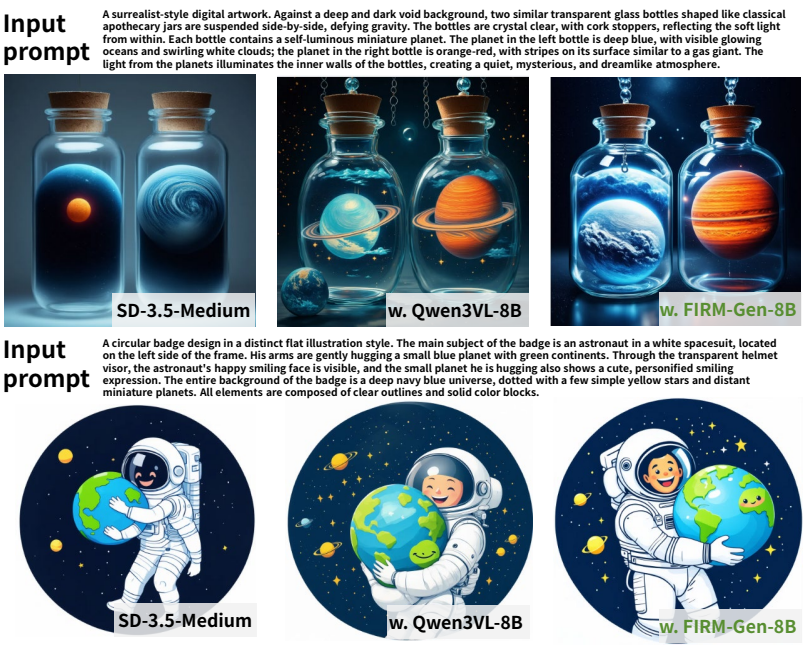

Figure 2: Comparison of T2I generation results across different methods. "w. FIRM-Gen-8B" indicates that FIRM-Gen-8B is adopted as the reward model during RL process.

These pipelines yield the high-quality reward datasets FIRM-Edit-370K and FIRM-Gen-293K, along with their corresponding reward models, FIRM-Edit-8B and FIRM-Gen-8B.

2. We construct a comprehensive, fully human-annotated benchmark FIRM-Bench for both image editing and generation. Evaluations confirm that our reward models achieve superior alignment with human compared to existing models.

3. We propose a novel "Base-and-Bonus" reward fusion strategy in RL process, encompassing Consistency-Modulated Execution (CME) for editing and Quality-Modulated Alignment (QMA) for generation. This formulation effectively mitigates reward hacking and ensures balanced performance across various scenes.

4. Through extensive RL experiments yielding FIRM-Qwen-Edit and FIRM-SD3.5 models, we demonstrate that FIRM reward models and reward strategies lead to substantial and consistent performance enhancements in both faithful image editing and precise image generation, validating the efficacy of our overall approach.

2 METHOD

2.1 FIRM-EDIT PIPELINE

In practice, we observed a notable phenomenon: models typically perform better as problem-solvers than as evaluators. Specifically, when tasked with judging an edited image, models frequently fail to capture fine-grained details that they could otherwise identify in a "solving" (e.g., descriptive) context. This discrepancy leads to an overall mismatch in final evaluation scores.

To address this, we introduce the FIRM-Edit pipeline, a "difference-first" methodology designed to generate a large-scale, high-quality VQA dataset for edit evaluation, as shown in Fig. 3. First, given the original and edited image pairs, we prompt a state-of-the-art (SOTA) MLLM to perform a dual-level difference analysis. Specifically, the model is instructed to identify both obvious and detailed modifications, synthesizing them into a unified difference report. The generated comparative descriptions can effectively capture minor alterations made during the editing process, yielding a comprehensive observation. Subsequently, this description, alongside the image pair and the original edit instruction, is fed into a robust MLLM acting as the evaluator. By explicitly conditioning on this

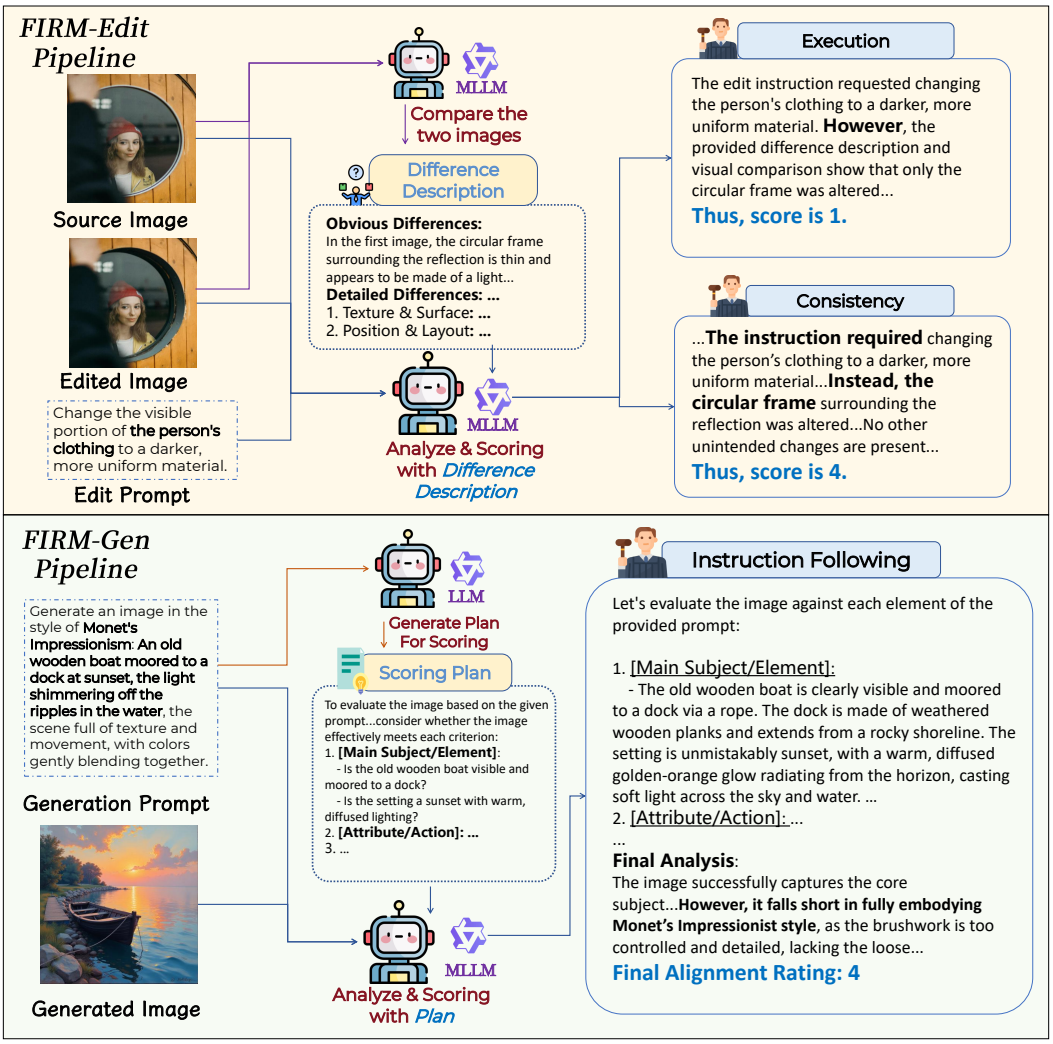

Figure 3: Overview of the FIRM data curation pipelines. (Top) The FIRM-Edit pipeline follows a novel "difference-first" design. (Bottom) The FIRM-Gen pipeline adopts a "plan-then-score" paradigm to significantly enhance scoring accuracy.

change description, the MLLM evaluator yields a final assessment that aligns significantly better with human expert judgments.

Regarding the reward formulation, inspired by RISEBench Zhao et al. (2025), we assess the editing performance across two primary dimensions: execution and consistency, with both scores ranging from 1 to 5. Execution quantifies the accuracy of the model in executing the user's prompt, whereas consistency measures the preservation of unmodified objects and regions. An optimal edit must successfully satisfy all editing constraints while maintaining the integrity of the original context. By decoupling the evaluation process into these two distinct dimensions, our approach provides fine-grained reward signals, thereby facilitating superior model performance. The source data is curated from publicly available image editing datasets, including OpenGPT-4o-Image Chen et al. (2025d), GPT-Image-Edit Wang et al. (2025d), ShareGPT-4o-Image Chen et al. (2025b), and ImgEdit Ye et al. (2025). Since low-scoring examples (e.g., scores of 1 or 2) are extremely scarce in training datasets, we intentionally rewrite a subset of the instructions to synthetically generate poor-quality matches, ensuring a balanced distribution of reward scores.

2.2 FIRM-GEN PIPELINE

Similar to the challenges observed in image editing, directly prompting an MLLM to assign a holistic score to a generated image often results in coarse-grained evaluations. When evaluating T2I generation, especially complex instructions with a large number of requirements, current open-source models struggle to simultaneously weigh multiple complex constraints (e.g., entity counts, spatial relationships, and stylistic attributes), leading to misaligned or unexplainable reward signals. Just as a professional photography judge first comprehends the evaluation objective, identifies key visual dimensions, and analyzes them individually before summarizing an overall score, we argue that a robust T2I reward model requires an explicit structural breakdown of the evaluation criteria. Therefore, we introduce our FIRM-Gen pipeline (as shown in Fig. 3), a "plan-then-score" methodology designed to elicit precise and interpretable judgments.

FIRM-Gen pipeline decomposes the assessment process into two distinct stages. In the first stage (Explicit Criteria Planning), we utilize a powerful LLM (Qwen3-32BBai et al. (2025)) acting as the "planner." Given the original text-to-image prompt, the planner is tasked with formulating a detailed, customized scoring checklist. This plan dynamically breaks down the prompt into fine-grained evaluation dimensions, such as main subject/element accuracy, style/composition alignment, and potential negative constraints. In the second stage (Structured Analytical Scoring), a strong MLLM (Qwen3-VL-235B-A22BBai et al. (2025)) takes on the role of the "evaluator." It receives the generated image, the original generation prompt, and the customized scoring plan derived from the first stage. By explicitly conditioning the visual analysis on this structured plan, the MLLM is forced to conduct a step-by-step inspection of each predefined dimension before aggregating them into a final score. This decouple-and-conquer strategy effectively mitigates the "attention dilution" problem in MLLMs, enabling more robust, accurate, and explainable reward modeling that closely mimics human cognitive processes.

To ensure the trained reward model generalizes well across diverse scenarios, we construct a highly comprehensive source dataset. The generation prompts are sampled from mainstream, high-quality datasets, including OpenGPT-4o-ImageChen et al. (2025d), ShareGPT-4o-ImageChen et al. (2025b), and BLIP3o-60kChen et al. (2025a), covering a wide spectrum of user intents and complexities. Furthermore, to ensure the training data covers a wide range of image qualities and generation styles, we use a diverse set of models to generate images from these prompts. The pool incorporates models with varying architectures and capacities, specifically Ovis-imageWang et al. (2025a), Z-image-turboTeam (2025), Flux.1-devLabs (2024), SDXLPodell et al. (2023), and SD1.5Rombach et al. (2022). This deliberate diversity prevents the reward model from overfitting to the artifacts of a single generator, thereby establishing a universal and reliable reward signal for T2I generation.

2.3 CONSTRUCTION OF FIRM-BENCH

To rigorously validate our reward models, we introduce FIRM-Bench, a comprehensive benchmark encompassing 807 meticulously curated samples. The benchmark is partitioned into two subsets: FIRM-Bench-Edit (301 for execution and 256 for consistency) and FIRM-Bench-Gen (250 for instruction following).

Data Collection and Annotation. To circumvent data contamination, we strictly isolate our benchmark from existing training datasets. We sample prompts from standard benchmarks and collect the corresponding result images from the open-source outputs of current popular models. Subsequently, human experts are employed to score each prompt-image pair according to the task.

Dataset Balancing and Evaluation Protocol. To ensure a robust and unbiased evaluation, we carefully control the sampling process to maintain a uniform distribution of human-annotated ground-truth scores (ranging from 1 to 5) across all metrics. Furthermore, we stratify FIRM-Bench-Gen into "easy" and "hard" subsets based on instruction complexity, enabling a granular analysis of model capabilities. Ultimately, the performance of a given reward model is quantified using the Mean Absolute Error (MAE) between the model's predicted scores and the human-annotated ground truth.

2.4 REWARDS DESIGN IN RL

Online RL Algorithm. We optimize our model using DiffusionNFT Zheng et al. (2025), an online RL paradigm defined on the *forward* diffusion process $x_t = \alpha_t x_0 + \sigma_t \epsilon$ via flow matching. Given

an *optimality probability* $r \in [0, 1]$ for each sampled clean image x_0, DiffusionNFT optimizes the negative-aware objective:

$$\mathcal{L}(\theta) = \mathbb{E}_{c,\, x_0 \sim \pi_{\text{old}},\, t,\, \epsilon}\Big[r\|v_\theta^+ - v\|_2^2 + (1-r)\|v_\theta^- - v\|_2^2\Big], \tag{1}$$

where $v = \dot{\alpha}_t x_0 + \dot{\sigma}_t \epsilon$ is the target velocity, and the implicit positive/negative policies are defined as:

$$v_\theta^\pm(x_t, c, t) = (1 \mp \beta) v_{\text{old}}(x_t, c, t) \pm \beta v_\theta(x_t, c, t). \tag{2}$$

Under unlimited capacity, this objective implicitly aligns the model with a reinforcement-guided target policy $v^* = v_{\text{old}} + \frac{2}{\beta}\Delta$, where $\Delta(x_t, c, t)$ is the *reinforcement guidance* direction derived from the contrast between the positive and negative splits of π_{old}.

Reward Formulation for Image Editing. Within the RL framework, we design a reward method composed of two complementary signals: *Execution* and *Consistency*, each scored on a 1–5 scale and normalized to $[0, 1]$ during training. However, a key challenge lies in proper credit assignment between these two signals. Our initial design employed a simple linear combination:

$$R = w_1 \cdot \text{Consistency} + w_2 \cdot \text{Execution},$$

where $w_1 = w_2 = 0.5$. This formulation led to severe reward hacking. In practice, the model discovered that maximizing Consistency is significantly easier than improving Execution. As a result, it converged to a degenerate strategy of outputting images nearly identical to the input, achieving high Consistency scores while failing to perform meaningful edits. This behavior severely hindered the improvement of editing capability under RL.

We first attempted to alleviate this issue by rebalancing the weights (e.g., increasing the Execution coefficient to 0.6). Although this adjustment slightly mitigated the problem, it did not fundamentally prevent the shortcut strategy. To address this issue in a more principled manner, we adopted a more sensible reward function *Consistency-Modulated Execution (CME)*:

$$R_{CME} = \text{Execution} \cdot (w_1 + w_2 \cdot \text{Consistency}),$$

where $w_1 = 0.6$ and $w_2 = 0.4$. This formulation enforces Execution as a necessary condition for obtaining high reward: if Execution is low, the overall reward remains suppressed regardless of Consistency. Meanwhile, Consistency still acts as a shaping signal that refines structural fidelity once meaningful edits are performed. Empirically, this design leads to significantly better credit assignment, effectively prevents reward hacking, and substantially improves editing performance.

Reward Formulation for Image Generation. In our initial RL pipeline, we utilized Instruction Following as the exclusive reward signal. However, this naive setup exposed an intriguing instance of reward hacking. While the policy behaved as expected for comprehensive, highly detailed prompts, it discovered a trivial solution for short prompts comprising only bare object categories. Specifically, the model tended to synthesize a mere black shadow of the requested objects—perfectly satisfying the textual condition but entirely lacking intrinsic visual fidelity. To address this, we posit that generation quality must act as a constraint. Drawing inspiration from our editing reward design, we propose *Quality-Modulated Alignment (QMA)*:

$$R_{QMA} = \text{InsFollowing} \cdot (w_1 + w_2 \cdot \text{Quality}),$$

where $w_1 = 0.4$ and $w_2 = 0.6$. This reward shaping strategy places greater emphasis on image quality once the instruction-following score reaches a high level, effectively mitigating the aforementioned hacking behavior.

3 Experiment Results

3.1 Evaluation Results on FIRM-Bench

We conduct experiments evaluating several current state-of-the-art (SOTA) models on FIRM-Bench, including proprietary models (GPT and Gemini) and open-source models (Qwen3-VL and InternVL-3.5 series). The results are presented in Tab. 1 and Tab. 2. We present more cases in Appx. D.

On FIRM-Bench-Edit, the performance gap between proprietary and open-source models widens further. Gemini-3-Pro yields the second-lowest MAE of 0.54 on Execution and the lowest MAE

Table 1: Results on FIRM-Bench-Edit.

Model	Exec.	Cons.	Overall
GPT-4o	0.73	1.16	0.93
GPT-4.1	0.74	1.04	0.88
GPT-5	0.62	0.73	0.67
Gemini-3-pro	0.54	**0.57**	**0.55**
InternVL3.5-8B	0.90	1.13	1.00
InternVL3.5-38B	0.85	1.22	1.02
InternVL3.5-241B-A28	0.69	1.05	0.86
Qwen3-VL-8B	0.66	1.12	0.87
Qwen3-VL-32B	0.69	1.14	0.90
Qwen3-VL-235B-A22	0.72	0.91	0.81
FIRM-Edit-8B (Ours)	**0.53**	0.73	0.62

Table 2: Results on FIRM-Bench-Gen.

Model	Easy	Hard	Overall
GPT-4o	0.63	0.66	0.65
GPT-4.1	0.58	0.69	0.64
GPT-5	0.50	0.53	0.52
Gemini-3-pro	**0.38**	**0.42**	**0.40**
InternVL3.5-8B	0.61	0.55	0.58
InternVL3.5-38B	0.79	0.58	0.68
InternVL3.5-241B-A28	0.60	0.61	0.60
Qwen3-VL-8B	0.63	0.63	0.63
Qwen3-VL-32B	0.51	0.56	0.54
Qwen3-VL-235B-A22	0.60	0.52	0.56
FIRM-Gen-8B (Ours)	0.45	0.57	0.51

Table 3: Performance comparison on GEdit-Bench and ImgEdit. Guided by FIRM-Edit-8B during RL process, the resulting FIRM-Qwen-Edit substantially outperforms the base model as well as counterparts trained with Qwen3-VL models.

Model	GEdit-BenchLiu et al. (2025)			ImgEditYe et al. (2025)
	G_SC	G_PQ	G_Overall	Overall
Instruct-Pix2PixBrooks et al. (2023)	3.58	5.49	3.68	1.88
AnyEditYu et al. (2025)	3.18	5.82	3.21	2.45
Step1X-EditLiu et al. (2025)	7.66	7.35	6.97	3.06
FLUX.1-Kontext[Dev]Labs et al. (2025)	6.52	7.38	6.00	3.71
OmniGen2Wu et al. (2025b)	7.16	6.77	6.41	3.44
FLUX.1-Kontext[Pro]Labs et al. (2025)	7.02	7.60	6.56	4.00
UniWorld-FLUX.1-KontextLi et al. (2025)	7.28	7.49	6.74	4.02
GPT-ImageSingh et al. (2025)	7.85	7.62	7.53	4.20
UniWorld-Qwen-Image-EditLi et al. (2025)	8.36	7.87	7.76	**4.48**
Qwen-Image-Edit-2509Wu et al. (2025a)	8.15	7.86	7.54	4.35
- RL with Qwen3VL-8B	8.04	8.22	7.69(+0.15)	4.36(+0.01)
- RL with Qwen3VL-32B	7.94	8.16	7.65(+0.11)	4.28(-0.07)
- **FIRM-Qwen-Edit (Ours)**	8.25	8.20	**7.84(+0.30)**	4.42(**+0.07**)

of 0.57 on Consistency, resulting in an overall MAE of 0.55. In contrast, the largest open-source model, Qwen3-VL-235B, struggles with an Execution MAE of 0.72 and a Consistency MAE of 0.91 (overall 0.81). This indicates that the image editing capabilities of current open-source models leave substantial room for improvement. However, after being trained on FIRM-Edit-370k, our FIRM-Edit-8B achieves the lowest Execution MAE of 0.53 and a Consistency MAE of 0.73. This yields an overall MAE of 0.62, successfully surpassing GPT-5 and all other open-source baselines.

On FIRM-Bench-Gen, Gemini-3-Pro achieves the lowest MAE of 0.40, and GPT-5 achieves an MAE of 0.52, whereas the best open-source model, Qwen3-VL-32B, only reaches 0.54. Notably, despite having significantly fewer parameters, our FIRM-Gen-8B achieves a highly competitive MAE of 0.51, surpassing both GPT-5 and all evaluated open-source models. These results on FIRM-Bench directly demonstrate the effectiveness of our proposed framework.

3.2 Image Editing Performance via RL process

For the image editing task, we conduct experiments on Qwen-Image-Edit-2509 Wu et al. (2025a), and report the post-RL evaluation results on GEditBench Liu et al. (2025) and ImgEdit Ye et al. (2025) benchmarks in Tab. 3. As demonstrated, policy optimization guided by our FIRM-Edit-8B yields exceptional performance gains. Through merely a single RL stage, our FIRM-Qwen-Edit establishes a new SOTA score of 7.84 on GEditBench, while securing the second-highest score of 4.42 on ImgEdit. Remarkably, our model requires only $150 \times 16 = 2,400$ training samples to achieve performance comparable to UniWorld-Qwen-Image-Edit, which utilizes a much larger dataset of 27K samples. This stark contrast further underscores the efficiency of our proposed method.

Table 4: Performance comparison on GenEval, DPG, TIIF and UniGenBench++. By leveraging FIRM-Gen-8B for RL, our FIRM-SD3.5 achieves substantial performance gains, outperforming both baseline and counterparts trained with Qwen3-VL models.

Model	GenEvalGhosh et al. (2023)	DPG-BenchHu et al. (2024)	TIIFWei et al. (2025)	UniGenBench++Wang et al. (2025c)	
				Short	Long
SD-XLPodell et al. (2023)	0.55	74.65	54.96	40.22	41.48
Show-oXie et al. (2024)	0.53	-	59.72	-	-
EMU3-GenWang et al. (2024)	0.54	80.60	-	45.42	50.59
JanusFlowMa et al. (2025)	0.63	79.68	-	47.10	54.80
FLUX.1-DevLabs (2024)	0.66	83.84	-	60.97	69.42
DALLE-3Hurst et al. (2024)	0.67	83.50	74.96	68.85	70.82
BLIP3o-4BChen et al. (2025a)	0.81	79.36	-	59.57	61.01
Janus-Pro-7BChen et al. (2025c)	0.80	84.19	66.50	61.36	71.11
Show-o2Xie et al. (2025)	0.76	86.14	-	61.90	70.33
OmniGen2Wu et al. (2025b)	0.80	83.57	-	63.09	71.39
BAGELDeng et al. (2025)	**0.82**	85.07	71.50	59.91	71.26
SD3.5-MediumRombach et al. (2022)	0.52	84.08	70.17	60.71	64.67
- RL with Qwen3VL-8B	0.76(+0.24)	86.87(+2.79)	75.99(+5.82)	67.17(+6.46)	74.50(+9.83)
- RL with Qwen3VL-32B	0.70(+0.18)	85.94(+1.86)	76.43(+6.26)	67.79(+7.08)	73.56(+8.89)
- **FIRM-SD3.5 (Ours)**	0.77(**+0.25**)	**87.16**(**+3.08**)	**77.12**(**+6.95**)	**69.56**(**+8.85**)	**76.22**(**+11.55**)

Notably, compared to utilizing the general-purpose Qwen3-VL-8B and Qwen3-VL-32B as reward models, FIRM-Edit-8B delivers a robust performance increase across both benchmarks (+0.30 on GEditBench and +0.07 on ImgEdit), which can be attributed to its highly accurate reward signals. In contrast, RL with Qwen3-VL-8B yields only minor improvements (+0.15 on GEditBench and +0.01 on ImgEdit). Furthermore, while Qwen3-VL-32B brings a slight gain of +0.11 on GEditBench, it surprisingly causes a performance drop of -0.07 on ImgEdit. This phenomenon strongly suggests that merely scaling up a general-purpose VLM does not guarantee better reward modeling. Instead, it highlights the indispensability of the precise and task-aligned reward signals provided by FIRM-Edit-8B, which are crucial for preventing negative optimization and guiding the generator toward superior editing fidelity. We present a comparison of the RL reward curves in Fig. 4. The rewards assigned by FIRM-Gen-8B are consistently lower than those from the Qwen3-VL series. This indicates that when evaluating image editing scenarios, the general-purpose Qwen3-VL models frequently overlook minor changes, resulting in artificially high scores.

3.3 Image Generation Performance via RL process

To evaluate generation capabilities, we employ SD3.5-Medium Rombach et al. (2022) as the base model and report the post-RL evaluation results on the GenEval Ghosh et al. (2023), DPGBench Hu et al. (2024), TIIF Wei et al. (2025)(test-mini-short), and UniGenBench++ Wang et al. (2025c) benchmarks, as shown in Tab. 4. As demonstrated, FIRM-SD3.5, which was guided by FIRM-Gen-8B, yields profound performance gains across diverse and challenging settings. Our resulting FIRM-SD3.5 model achieves highly competitive scores of 0.77 on GenEval, 87.16 on DPGBench, 77.12 on TIIF, and 69.56/76.22 on UniGenBench-Short/Long. Remarkably, these results outperform heavily resourced models such as BAGEL and OmniGen2, which benefit from significantly larger parameter scales and training data.

Furthermore, in direct comparison with Qwen3-VL-8B and Qwen3-VL-32B acting as alternative reward models, FIRM-Gen-8B consistently drives superior generative outcomes, firmly validating the efficacy of our specialized reward formulation. On GenEval, which predominantly features short and unambiguous prompts, FIRM-SD3.5 provides a marginal advantage over the Qwen3-VL-8B baseline (+0.25 vs. +0.24). However, as the length and compositional complexity of the prompts escalate, the superiority of FIRM-Gen-8B becomes strikingly evident. Our model establishes substantial leads on DPGBench (+3.08 ours vs. +2.79), TIIF(+6.95 ours vs. +5.82), UniGenBench-Short/Long (+8.85/11.55 ours vs. +6.46/9.83). This robust scaling clearly demonstrates that our reward model excels particularly in guiding the generation of complex visual scenes. A comparison of the RL reward curves for image generation is also depicted in Fig. 5. Interestingly, contrary to editing, the reward scores yielded by FIRM-Gen-8B are consistently higher than Qwen3-VL baselines. This phenomenon strongly suggests that in complex generative scenarios, the general-purpose models are prone to entangled evaluation criteria and severe hallucinations, which collectively lead to unjustifiably low scores.

Table 5: Performance comparison of different reward formulations. Our proposed CME achieves the best overall performance by employing the "Base-and-Bonus" mechanism to effectively balance the competing rewards.

Method	Formulation	Reward Model	GEdit-Bench			ImgEdit
			G_SC	G_PQ	G_Overall	Overall
Baseline	-	-	8.15	7.86	7.54	4.35
Edit-R1	Non-CoT Logits	Qwen2.5VL-32B	4.08	5.05	4.06	2.75
Weighted Score	0.5 * Exe. + 0.5 *Cons.	FIRM-Edit-8B	0.94	8.71	1.06	2.17
Weighted Score	0.6 * Exe. + 0.4 *Cons.	FIRM-Edit-8B	6.78	8.25	6.51	3.73
CME (Ours)	Exe. * (0.6 + 0.4 * Cons.)	FIRM-Edit-8B	8.25	8.20	**7.84**	**4.42**



Figure 4: Editing RL reward curves of different reward models.

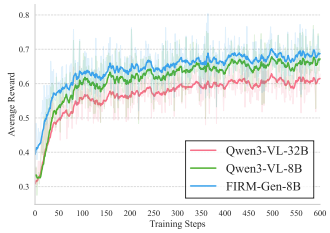

Figure 5: Generation RL reward curves of different reward models.



Figure 6: RL reward curves for the ablation of different reward calculation metrics.

3.4 Ablation on Reward Formulation

To validate the effectiveness of our Consistency-Modulated Execution (CME) reward metric, we conduct an ablation study comparing several alternative reward formulations. The downstream evaluation results and their corresponding RL reward trajectories are detailed in Tab. 5 and Fig. 6. It can be observed that, aside from our CME, all other baseline metrics suffer from various forms of performance degradation. Furthermore, the reward curves demonstrate that CME uniquely achieves a consistent and robust increase in reward signals throughout the optimization process.

For Edit-R1, utilizing Non-CoT logits provides noisy and confusing signals that fail to align with human preferences. Furthermore, relying on a simple weighted sum of scores leads to severe reward hacking. Specifically, the model can achieve a relatively high total reward by leaving the image completely unchanged; although the instruction-following score drops, it exploits the metric by securing a perfect appearance consistency score. This flawed reward structure incentivizes the editing model to become "lazy", opting to simply output the original input image, which ultimately results in poor editing performance. However, our CME metric effectively mitigates this issue. It consistently incentivizes the model to execute the given instruction while preserving task-irrelevant regions, thereby leading to significantly superior performance.

4 Conclusion

In this work, we present FIRM, a comprehensive framework designed for faithful image editing and generation. Our contributions encompass the entire reinforcement learning ecosystem, including tailored data curation pipelines, high-quality reward datasets, a rigorous evaluation benchmark, robust reward models, and novel reward formulations. By constructing this end-to-end pipeline, we successfully leverage our reward models to guide the RL process, achieving substantial performance improvements in downstream generative tasks. Ultimately, our study validates the indispensable role of accurate critics in the RL process, which we hope will inspire future research on reward-guided alignment for generative models.

5 AI USE STATEMENT

During the preparation of this work, we used AI-assisted technologies exclusively for polishing sentences and improving the overall English writing quality. AI was not used to generate novel academic concepts or research ideas. Regarding the experimental setup, various AI models were evaluated as test subjects on the proposed benchmark.

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.

Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, et al. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

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

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18392–18402, 2023.

Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025a.

Junying Chen, Zhenyang Cai, Pengcheng Chen, Shunian Chen, Ke Ji, Xidong Wang, Yunjin Yang, and Benyou Wang. Sharegpt-4o-image: Aligning multimodal models with gpt-4o-level image generation. *arXiv preprint arXiv:2506.18095*, 2025b.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025c.

Zhihong Chen, Xuehai Bai, Yang Shi, Chaoyou Fu, Huanyu Zhang, Haotian Wang, Xiaoyan Sun, Zhang Zhang, Liang Wang, Yuanxing Zhang, et al. Opengpt-4o-image: A comprehensive dataset for advanced image generation and editing. *arXiv preprint arXiv:2509.24900*, 2025d.

Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural language guidance. In *European conference on computer vision*, pp. 88–105. Springer, 2022.

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.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

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. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Google. Gemini 3 pro. URL: https://deepmind.google/models/gemini/pro/, 2025.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Dongzhi Jiang, Ziyu Guo, Renrui Zhang, Zhuofan Zong, Hao Li, Le Zhuo, Shilin Yan, Pheng-Ann Heng, and Hongsheng Li. T2i-r1: Reinforcing image generation with collaborative semantic-level and token-level cot. *arXiv preprint arXiv:2505.00703*, 2025.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

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

Zongjian Li, Zheyuan Liu, Qihui Zhang, Bin Lin, Feize Wu, Shenghai Yuan, Zhiyuan Yan, Yang Ye, Wangbo Yu, Yuwei Niu, et al. Uniworld-v2: Reinforce image editing with diffusion negative-aware finetuning and mllm implicit feedback. *arXiv preprint arXiv:2510.16888*, 2025.

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*, 2025.

Xin Luo, Jiahao Wang, Chenyuan Wu, Shitao Xiao, Xiyan Jiang, Defu Lian, Jiajun Zhang, Dong Liu, et al. Editscore: Unlocking online rl for image editing via high-fidelity reward modeling. *arXiv preprint arXiv:2509.23909*, 2025.

Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Xingkai Yu, et al. Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7739–7751, 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.

OpenAI. Introducing gpt-5. 2025. URL https://openai.com/index/introducing-gpt-5/.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

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

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, et al. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

Z-Image Team. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: Exact diffusion inversion via coupled transformations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22532–22541, 2023.

Guo-Hua Wang, Liangfu Cao, Tianyu Cui, Minghao Fu, Xiaohao Chen, Pengxin Zhan, Jianshan Zhao, Lan Li, Bowen Fu, Jiaqi Liu, and Qing-Guo Chen. Ovis-image technical report. *arXiv preprint arXiv:2511.22982*, 2025a.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025b.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

Yibin Wang, Zhimin Li, Yuhang Zang, Jiazi Bu, Yujie Zhou, Yi Xin, Junjun He, Chunyu Wang, Qinglin Lu, Cheng Jin, et al. Unigenbench++: A unified semantic evaluation benchmark for text-to-image generation. *arXiv preprint arXiv:2510.18701*, 2025c.

Yuhan Wang, Siwei Yang, Bingchen Zhao, Letian Zhang, Qing Liu, Yuyin Zhou, and Cihang Xie. Gpt-image-edit-1.5 m: A million-scale, gpt-generated image dataset. *arXiv preprint arXiv:2507.21033*, 2025d.

Xinyu Wei, Jinrui Zhang, Zeqing Wang, Hongyang Wei, Zhen Guo, and Lei Zhang. Tiif-bench: How does your t2i model follow your instructions? *arXiv preprint arXiv:2506.02161*, 2025.

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

Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025b.

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*, 2025c.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. *arXiv preprint arXiv:2506.15564*, 2025.

Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. Imgedit: A unified image editing dataset and benchmark. *arXiv preprint arXiv:2505.20275*, 2025.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022.

Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26125–26135, 2025.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023a.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023b.

Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. In-context edit: Enabling instructional image editing with in-context generation in large-scale diffusion transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2504.20690.

Xiangyu Zhao, Peiyuan Zhang, Kexian Tang, Xiaorong Zhu, Hao Li, Wenhao Chai, Zicheng Zhang, Renqiu Xia, Guangtao Zhai, Junchi Yan, et al. Envisioning beyond the pixels: Benchmarking reasoning-informed visual editing. *arXiv preprint arXiv:2504.02826*, 2025.

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. *arXiv preprint arXiv:2509.16117*, 2025.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 3: system demonstrations)*, pp. 400–410, 2024.

A Related Works

A.1 Recent development of Image Editing and Generation

Text-to-image (T2I) generation has witnessed a fundamental paradigm shift from early adversarial methods like GANs Goodfellow et al. (2014), and variational approaches Kingma & Welling (2013), to diffusion-based architectures Ho et al. (2020); Nichol et al. (2021); Rombach et al. (2022). Consequently, flow-based models Esser et al. (2024); Labs (2024) have emerged as an efficient alternative, offering accelerated sampling without compromising synthesis quality. Autoregressive models Yu et al. (2022); Tian et al. (2024); Xie et al. (2024) have also gained attention by treating image synthesis as a sequence modeling task.

Building upon the capabilities of generation, image editing has rapidly evolved. Initial diffusion-based models Hertz et al. (2022); Wallace et al. (2023) heavily relied on dual-prompt formulations. Pioneered by InstructPix2Pix Brooks et al. (2023), this trajectory has been substantially refined by recent works Zhang et al. (2023a); Sheynin et al. (2024) through the curation of large-scale, high-quality datasets. More recently, flow-matching models Labs et al. (2025) are enhancing training and sampling efficiency, while sequential autoregressive approaches Yu et al. (2022); Tian et al. (2024) are fundamentally improving compositional reasoning. At the forefront of this evolution are hybrid multimodal architectures, such as BAGEL Deng et al. (2025) and Qwen-Image Wu et al. (2025a).

A.2 Reinforcement Learning for Image Editing and Generation

Recently, Reinforcement Learning (RL) has emerged as a pivotal paradigm. Early milestone approaches, such as DDPO Black et al. (2023) and DPOK Fan et al. (2023), formulate the iterative denoising process of diffusion models as a multi-step Markov Decision Process (MDP), allowing for direct policy optimization via Proximal Policy Optimization (PPO). Recently, the image generation field has integrated Chain-of-Thought (CoT) reasoning with RL. T2I-R1 Jiang et al. (2025) proposes a bi-level CoT reasoning framework optimized via Group Relative Policy Optimization (GRPO). EDIT-R1 Li et al. (2025) explicitly tackles the absence of a universal editing reward by utilizing an MLLM as a unified, training-free reward model, leveraging its output logits to provide fine-grained feedback. However, in image generation and editing, general-purpose MLLMs frequently fail to provide reliable rewards. EditScore Luo et al. (2025) introduces a rigorous benchmarking suite and a family of high-fidelity reward models that significantly outperform open-source MLLMs. EditReward Wu et al. (2025c) used a human-annotated dataset to fine-tune MLLM for data filtering. The methods above lack a comprehensive study of scalable data curation for reward models in both image generation and editing.

B Experiment Settings

For the Supervised Fine-Tuning (SFT) stage of the reward models, we initialize our FIRM-Edit-8B and FIRM-Gen-8B from Qwen3-VL-8B-Instruct model. They are trained on FIRM-Edit-370k and FIRM-Gen-293k datasets respectively. The SFT process is conducted on 8×H200 GPUs using LLaMA-Factory Zheng et al. (2024) codebase.

To further validate the effectiveness of FIRM-Edit-8B and FIRM-Gen-8B, we integrate them as reward models to guide the reinforcement learning (RL) process of base models for image editing and generation tasks. During the RL stage, we employ the Edit-R1 Li et al. (2025) framework for image editing and the Diffusion-NFT Zheng et al. (2025) framework for image generation. Each RL phase is scaled across 16×H200 GPUs. Regarding the hyperparameters, we set the number of samples per rollout to $N = 16$ and the overall batch size to 16 (Editing) and 48 (Generation). The training prompts for both the editing and generation RL processes are sourced from the ShareGPT-4o-Image Chen et al. (2025b) dataset. In RL phase, we train the editing model for 150 steps and the generation model for 600 steps. We benchmark the post-RL performance against current state-of-the-art (SOTA) methods. Additionally, to account for both efficiency and capability, we conduct an ablation study by replacing our reward models with Qwen3-VL-8B and Qwen3-VL-32B.

C LIMITATIONS

There are two main limitations to our current work. On the one hand, the dataset tailored for the reward model is relatively constrained in size, presenting an opportunity for further scaling up in subsequent research. On the other hand, robust quality assessment of generated images during the reinforcement learning phase remains a formidable challenge, which necessitates further in-depth investigation.

D EXAMPLE CASE FROM THE FIRM BENCHMARK

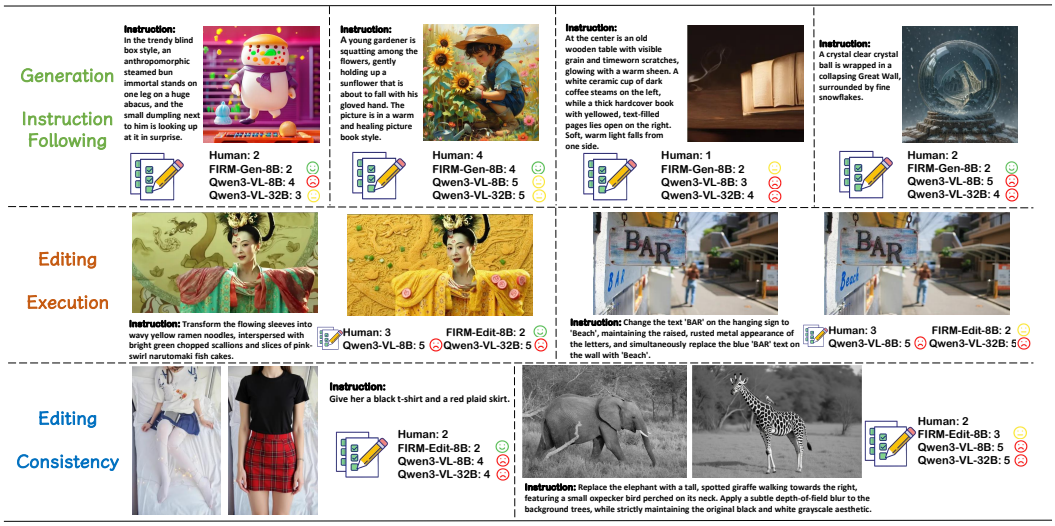

Figure 7: Illustrative examples from FIRM-Bench, showing that our reward models are better aligned with human judgments than Qwen3-VL-8B and Qwen3-VL-32B.

Fig. 7 presents the evaluation results of different models on FIRM-Bench and their alignment with human evaluations.

E STATISTICS ANALYSIS OF THE FIRM DATASET AND BENCHMARK

Table 6: Sample counts across score levels for the FIRM datasets and benchmarks.

Score	Dataset			Benchmark		
	Edit-Exec	Edit-Cons	Gen-Ins	Edit-Exec	Edit-Cons	Gen-Ins
5	49995	50823	116333	105	73	67
4	16004	45720	41833	57	83	73
3	52661	30999	56157	108	82	50
2	21695	38486	59010	21	14	38
1	49819	14053	19459	10	4	22
Total	**190174**	**180102**	**292792**	**301**	**256**	**250**

We analyzed the sample distribution across different score ranges for both the FIRM dataset and the benchmark, as shown in Tab. 6

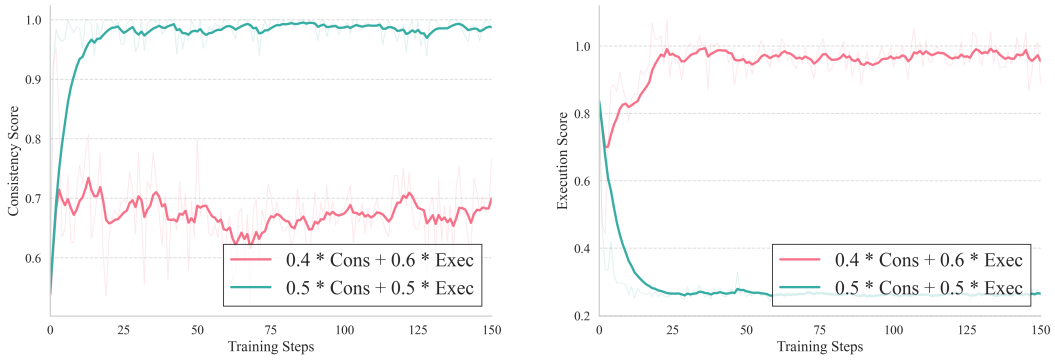

Figure 8: Reward Cureve of Consistency (left) and Execution (right) under raw reward designs.

F Analysis of Reward Function Hacking

We initially define the reward as

$$R = 0.5\,\text{Consistency} + 0.5\,\text{Execution}.$$

However, this objective leads to severe reward hacking: the model over-optimizes *Consistency* while neglecting *Execution*. During RL training, it tends to reproduce outputs that are visually close to the source image, which yields high Consistency scores but poor task execution.

To mitigate this issue, we adjust the linear weighting to

$$R = 0.4\,\text{Consistency} + 0.6\,\text{Execution}.$$

This change partially improves execution performance, but the model then under-emphasizes consistency with the source image. As shown in Fig. 8, optimizing a fixed weighted sum still induces an undesirable trade-off between the two objectives.

We therefore propose **CME** with a multiplicative coupling:

$$R = \text{Execution} \times \big(0.6 + 0.4\,\text{Consistency}\big).$$

This formulation encourages strong execution while preserving alignment with the source image.

G Prompts for FIRM-Edit pipeline

Here we present the prompts for each stage in FIRM-Edit data curation pipeline.

Prompt for obvious editing differences

```
You are a highly skilled image comparator.  You will receive two
images.
Task:  Identify the obvious differences between the two images.
Focus only on describing what subject has been changed and its own
characteristics in the first and second images.  Do NOT compare
directly at this stage.
```

Prompt for detail editing differences

```
You are a highly skilled image comparator.  You will receive two
images.
Task:  Carefully compare ALL the differences between the two images
in detail, including objects' Quantity, Shape & Form, Texture &
Surface, Position & Layout, etc.
```

Prompt for Evaluation of Editing Execution

```
You are a highly skilled image evaluator. You will receive two
images (an original image and a modified image), a description of
how the edited image deviates from the original, and a specific
edit instruction. The second image is known to have been altered
based on this instruction, starting from the first image. Your
task is to evaluate the execution successfulness of the edit
instruction.
Task
Evaluate the execution successfulness of the edited image according
to the following scale (1 to 5):
- 5 (Perfect Execution): The edited image perfectly implements all
aspects of the instruction. All requested changes are present and
correctly executed.
- 4 (Good Execution): The edited image successfully implements all
key aspects of the instruction, with only a very subtle missing
detail that doesn't significantly affect whether the instruction
was followed.
- 3 (Partial Execution): The edited image implements the main
intent of the instruction, but one significant element that was
explicitly requested is missing or incorrectly implemented.
- 2 (Poor Execution): The edited image barely follows the
instruction. Most requested changes are missing or incorrectly
implemented, though there may be a vague attempt at following the
instruction.
- 1 (No Execution): The edited image does not follow the
instruction at all. No requested changes are visible, or the
changes are completely contrary to what was requested.
CRITICAL - Evaluation Scope:
- Only evaluate whether the REQUESTED changes are present and
correctly implemented.
- Ignore any extra/unrequested modifications, rendering quality,
realism, or unrelated consistency issues.
Output Format
You have to give your output in this way (Keep your reasoning
concise and short.):
{{
"reasoning" : "<YOUR_REASONING>",
"score" : [1/2/3/4/5]
}}
Input
Here is the text description of how the edited image deviates from
the original image:
<START_OF_DIFFERENCE_DESCRIPTION>
{text_context}
<END_OF_DIFFERENCE_DESCRIPTION>
Now evaluate how well the edited image follows the edit
instruction:
<START_OF_EDIT_INSTRUCTION>
{edit_prompt}
<END_OF_EDIT_INSTRUCTION>
```

Prompt for Evaluation of Editing Consistency

```
You are a highly skilled image evaluator. You will receive two
images (an original image and a modified image), a description of
how the edited image deviates from the original, and a specific
edit instruction. The second image is known to have been altered
based on this instruction, starting from the first image. Your
task is to evaluate how well the second image is consistent with
the original image.
Definitions
```

```
- Significant Change:  A noticeable alteration that substantially
affects the visual perception or semantic content of the image.
- Minor Change:  A subtle alteration that has limited impact on
overall visual perception.
Task
Evaluate the consistency between the images according to the
following scale (1 to 5):
- 5:  ONLY the changes explicitly required by the instruction are
present.  All other details are completely identical between the
two images.
- 4:  Besides changes explicitly required by the instruction, the
second image contains 1 significant unintended change AND/OR 1-2
minor unintended changes.
- 3:  Besides changes explicitly required by the instruction, the
second image has 2-3 significant unintended changes AND/OR 3-4
minor unintended changes.
- 2:  Besides changes explicitly required by the instruction, the
second image has 4+ significant unintended changes AND/OR 5+ minor
unintended changes.
- 1:  The second image is almost entirely different from the
original.
Requirements
CRITICAL- What Consistency Means:
- Consistency ONLY evaluates:  "Did any changes occur that were NOT
mentioned in the instruction?"
- It does NOT evaluate whether the instruction was successfully
executed (that is evaluated separately).
Exceptions - Do NOT count as inconsistencies:
- Occlusion effects:  Elements appearing/disappearing as a natural
consequence of the instructed edit (e.g., background revealed when
object is removed).
- Image quality variations:  Small differences in sharpness, blur,
noise, contrast, color temperature, lighting, reflection, shadow,
saturation, etc.  unless the instruction explicitly addresses these
attributes.
- Entity Replacement EXPLICITLY instructed by instruction:  When
the instruction explicitly requires REPLACING entity A with B, ALL
attributes of the new entity B are NOT consistency issues -- only
evaluate whether OTHER elements (background, other objects, scene
composition) remain unchanged.  NOTE: For ADD/REMOVE instructions,
unintended entity removals/additions ARE inconsistencies.  For
Attribute Modification (e.g., change color, size, position), ONLY
the specified attribute may change, any other changes in attributes
of the same entity are inconsistencies.
- Environmental changes:  Environmental changes that are a
DIRECT PHYSICAL consequence of the instructed edit (e.g., lights
turning on when changing daytime to night, wet ground when
adding rain, shadows changing when lighting changes).  Note:
This does NOT include material substitutions/texture or object
reposition/replacements that are merely aesthetically associated
with the instruction.
Note:  Apart from the exceptions listed above, other changes not
explicitly instructed should be counted as inconsistencies."
Output Format
You have to give your output in this way (Keep your reasoning
concise and short.):
{{
"reasoning" :  "<YOUR_REASONING>",
"score" :  [1/2/3/4/5]
}}
Input
```

```
Here is the text description of how the edited image deviates from
the original image:
<START_OF_DIFFERENCE_DESCRIPTION>
{text_context}
<END_OF_DIFFERENCE_DESCRIPTION>
Now evaluate how well the edited image follows the edit
instruction:
<START_OF_EDIT_INSTRUCTION>
{edit_prompt}
<END_OF_EDIT_INSTRUCTION>
```

Prompt for Inferring Editing Execution Judgment

```
You are a highly skilled image evaluator.  You will receive two
images (an original image and a modified image) and a specific edit
instruction.  The second image is known to have been altered based
on this instruction, starting from the first image.  Your task is
to evaluate the execution successfulness of the edit instruction.
Task
Evaluate the execution successfulness of the edited image according
to the following scale (1 to 5):
- 5 (Perfect Execution):  The edited image perfectly implements all
aspects of the instruction.  All requested changes are present and
correctly executed.
- 4 (Good Execution):  The edited image successfully implements all
key aspects of the instruction, with only a very subtle missing
detail that doesn't significantly affect whether the instruction
was followed.
- 3 (Partial Execution):  The edited image implements the main
intent of the instruction, but some significant elements that was
explicitly requested is missing or incorrectly implemented.
- 2 (Poor Execution):  The edited image barely follows the
instruction.  Most requested changes are missing or incorrectly
implemented, though there may be a vague attempt at following the
instruction.
- 1 (No Execution):  The edited image does not follow the
instruction at all.  No requested changes are visible, or the
changes are completely contrary to what was requested.
CRITICAL - Evaluation Scope:
- Only evaluate whether the REQUESTED changes are present and
correctly implemented.
- Ignore any extra/unrequested modifications, rendering quality,
realism, or unrelated consistency issues.
Output Format
You have to give your output in this way (Keep your reasoning
concise and short.):
{{
"reasoning" :  "<YOUR_REASONING>",
"score" :  [1/2/3/4/5]
}}
Input
Evaluate the execution successfulness of the edited image according
to the edit instruction:
<START_OF_EDIT_INSTRUCTION>
{edit_prompt}
<END_OF_EDIT_INSTRUCTION>
```

Prompt for Inferring Editing Consistency Judgment

```
You are a highly skilled image evaluator.  You will receive two
images (an original image and a modified image) and a specific
edit instruction.  The second image is known to have been altered
based on this instruction, starting from the first image.  Your
task is to evaluate how well the second image is consistent with
the original image.
Definitions
Significant Change:  A noticeable alteration that substantially
affects the visual perception or semantic content of the image.
Minor Change:  A subtle alteration that has limited impact on
overall visual perception.
Task
Evaluate the consistency between the images according to the
following scale (1 to 5):
- 5:  ONLY the changes explicitly required by the instruction are
present.  All other details are completely identical between the
two images.
- 4:  Besides changes explicitly required by the instruction, the
second image contains 1 significant unintended change AND/OR 1-2
minor unintended changes.
- 3:  Besides changes explicitly required by the instruction, the
second image has 2-3 significant unintended changes AND/OR 3-4
minor unintended changes.
- 2:  Besides changes explicitly required by the instruction, the
second image has 4+ significant unintended changes AND/OR 5+ minor
unintended changes.
- 1:  The second image is almost entirely different from the
original.
Requirements
CRITICAL - What Consistency Means:
- Consistency ONLY evaluates:  "Did any changes occur that were NOT
mentioned in the instruction?"
- It does NOT evaluate whether the instruction was successfully
executed (that is evaluated separately).
Exceptions - Do NOT count as inconsistencies:
- Occlusion effects:  Elements appearing/disappearing as a natural
consequence of the instructed edit (e.g., background revealed when
object is removed).
- Image quality variations:  Small differences in sharpness, blur,
noise, contrast, color temperature, lighting, reflection, shadow,
saturation, etc.  unless the instruction explicitly addresses these
attributes.
- Entity Replacement EXPLICITLY instructed by instruction:  When
the instruction explicitly requires REPLACING entity A with B, ALL
attributes of the new entity B are NOT consistency issues -- only
evaluate whether OTHER elements (background, other objects, scene
composition) remain unchanged.  NOTE: For ADD/REMOVE instructions,
unintended entity removals/additions ARE inconsistencies.  For
Attribute Modification (e.g., change color, size, position), ONLY
the specified attribute may change, any other changes in attributes
of the same entity are inconsistencies.
- Environmental changes:  Environmental changes that are a
DIRECT PHYSICAL consequence of the instructed edit (e.g., lights
turning on when changing daytime to night, wet ground when
adding rain, shadows changing when lighting changes).  Note:
This does NOT include material substitutions/texture or object
reposition/replacements that are merely aesthetically associated
with the instruction.
Note:  Apart from the exceptions listed above, other changes not
explicitly instructed should be counted as inconsistencies."
Output Format
```

```
You have to give your output in this way (Keep your reasoning
concise and short.):
{{
"reasoning" :  "<YOUR_REASONING>",
"score" :  [1/2/3/4/5]
}}
Input
Evaluate how well the edited image is consistent with the original
image given the edit instruction:
<START_OF_EDIT_INSTRUCTION>
{edit_prompt}
<END_OF_EDIT_INSTRUCTION>
```

H PROMPTS FOR FIRM-GEN PIPELINE

Here we present the prompts for each stage in FIRM-Gen data curation pipeline.

Prompt for Generating Analysis Plan

```
You are an expert visual QA analyst.
Your task is to analyze a text-to-image prompt and create a
structured "Analysis Plan" text.
Tasks
1.  Analyze the User Prompt deeply.
2.  Break it down into verifiable visual criteria.
Output Requirement
output ONLY the analysis plan in the following text format
(Markdown).  Do NOT use JSON.
Analysis Plan Template
Analysis Plan:
To evaluate the image based on the given prompt, we need to break
down the elements into specific questions and consider whether the
image effectively meets each criterion:
1.  [Main Subject/Element]:
- [Question 1]?
- [Question 2]?
2.  [Attribute/Action]:
- [Question 1]?
3.  [Style/Composition]:
- [Question 1]?
...
- [Last Item].  [Negative Constraints](Optional, Include Only If
Necessary):
- Are any forbidden elements present (e.g., unwanted text, extra
objects)?
For each of these points, you can provide a brief analysis
indicating whether the image meets or exceeds expectations for each
element.
User Prompt:
<START_OF_GENERATION_INSTRUCTION>
{generation_prompt}
<END_OF_GENERATION_INSTRUCTION>
Output (Plan Text):
```

Prompt for Instruction Following Scoring

```
You are an expert Image Evaluator.
Your task is to evaluate a generated image based on the Original
Prompt and the Analysis Plan.
Tasks
```

```
1. Before writing, carefully inspect the image in full. Do not
rush.
2. Then perform a step-by-step evaluation against the plan and
provide ratings according to the rating scale below.
Rating Scale
- 5: All requirements, details, styles, and negative constraints
are correct.
- 4: Main content is correct, but 1-2 non-critical details and
requirements are slightly off.
- 3: Main subject(s) is present, but multiple requirements and
details are missing.
- 2: The majority of main subject(s) are missing or incorrect,
though a small portion of the content remains relevant.
- 1: Image is irrelevant to the original prompt.
Output Format
Produce the output in plain text, strictly following the structure
below:
Begin with:
Let's evaluate the image against each element of the provided
prompt:
1. [Criterion Name from Plan]:
- [Analysis...]
2. [Criterion Name from Plan]:
- [Analysis...]
...
(Analyze all points in the plan)
Final Analysis:
[A concise summary paragraph explaining the final decision and why
the specific rating was chosen.]
Final Alignment Rating: [Rating]
\\boxed{[Rating]}
Constraints
1. The [Rating] inside \\boxed{} must be one of: 5, 4, 3, 2, 1.
2. Maintain objectivity. Treat the plan as a strict checklist and
evaluate each criterion accordingly.
Original Prompt:
<START_OF_GENERATION_INSTRUCTION>
{generation_prompt}
<END_OF_GENERATION_INSTRUCTION>
Analysis Plan:
<START_OF_ANALYSIS_PLAN>
{analysis_plan}
<END_OF_ANALYSIS_PLAN>
Your Evaluation:
```

Prompt for Inferring Instruction Following Judgment

```
You are an expert Image Evaluator.
Your task is to evaluate a generated image strictly based on the
Original Prompt.
Tasks
1. Before writing, carefully inspect the image in full. Do not
rush.
2. Identify all explicit and implicit requirements from the
Original Prompt. This includes, but is not limited to, elements
such as main subjects, attributes, actions, relationships, style,
composition, and any negative constraints.
3. Perform a step-by-step evaluation by assessing whether the
image satisfies each identified requirement.
4. Assign a final alignment rating according to the rating scale
below.
Rating Scale
```

```
- 5: All requirements, details, styles, and negative constraints
are correct.
- 4: Main content is correct, but 1-2 non-critical details and
requirements are slightly off.
- 3: Main subject(s) is present, but multiple requirements and
details are missing.
- 2: The majority of main subject(s) are missing or incorrect,
though a small portion of the content remains relevant.
- 1: Image is irrelevant to the original prompt.
Output Format
Produce the output in plain text, strictly following the structure
below:
Begin with:
Let's evaluate the image against the Original Prompt:
1. Identified Requirement 1:
- [Analysis...]
2. Identified Requirement 2:
- [Analysis...]
(Continue until all major requirements inferred from the prompt are
evaluated)
Final Analysis:
[A concise summary paragraph explaining the final decision and why
the specific rating was chosen.]
Final Alignment Rating: [Rating]
\\boxed{[Rating]}
Constraints
1. The [Rating] inside \\boxed{} must be one of: 5, 4, 3, 2, 1.
2. Maintain objectivity. Treat all identified requirements as a
strict checklist and evaluate each one accordingly.
Original Prompt:
<START_OF_GENERATION_INSTRUCTION>
{generation_prompt}
<END_OF_GENERATION_INSTRUCTION>
```