RationalRewards: Reasoning Rewards Scale Visual Generation Both Training and Test Time

Haozhe Wang[1] **Cong Wei**[2] **Weiming Ren**[2] **Jiaming Liu**[3] **Fangzhen Lin**[1] **Wenhu Chen**[2]

[1] HKUST [2] University of Waterloo [3] Alibaba

Abstract

Most reward models for visual generation reduce rich human judgments to a single unexplained score, discarding the reasoning that underlies preference. We show that teaching reward models to produce explicit, multi-dimensional critiques before scoring transforms them from passive evaluators into active optimization tools—improving generators in two complementary ways: at training time, structured rationales provide interpretable, fine-grained rewards for reinforcement learning; at test time, a Generate–Critique–Refine loop turns critiques into targeted prompt revisions that improve outputs without any parameter updates. To train such a model without costly rationale annotations, we introduce Preference-Anchored Rationalization (PARROT), a principled framework that recovers high-quality rationales from readily available preference data through anchored generation, consistency filtering, and distillation. The resulting model, **RationalRewards** (8B), achieves state-of-the-art preference prediction among open-source reward models—competitive with Gemini-2.5-Pro—while using 10–20× less training data than comparable baselines. As an RL reward, it consistently improves text-to-image and image-editing generators beyond scalar alternatives. Most strikingly, its test-time critique-and-refine loop matches or exceeds RL-based fine-tuning on several benchmarks, suggesting that structured reasoning can unlock latent capabilities in existing generators that suboptimal prompts fail to elicit. Models and Code are available at **Project Page**.

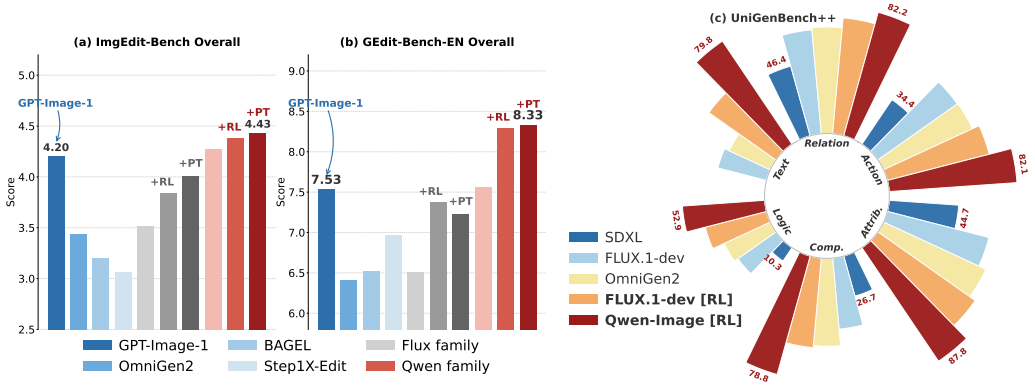
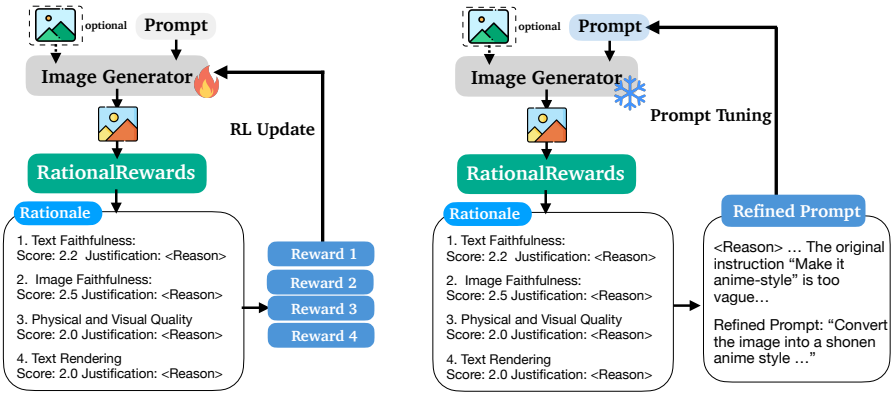


Figure 1: Train-Time RL and Test-Time PromptTuning (PT) with `RationalRewards` on text and image-to-image generation benchmarks. (Left) Comparison on image editing benchmarks. RL with `RationalRewards` outperforms prior open-source generators. Crucially, we find that test-time PT with `RationalRewards` alone can surpass expensive RL. (Right) Breakdown results on text-to-image benchmark UniGenBench++.

1 Introduction

As visual generation advances toward photorealistic, instruction-following outputs (Google DeepMind, 2025; OpenAI, 2025; Wu et al., 2025b; Esser et al., 2024), reward models that *eval-*



(a) Training Time RL (b) Test Time Prompt Tuning

Figure 2: `RationalRewards` is a reasoning-based reward model that produces structured rationales before assigning scores, enabling dual-space optimization for image generation. (a) As a reward model, it improves RL-based fine-tuning of generators over scalar baselines; (b) as a test-time optimizer, its Generate–Critique–Refine loop matches or surpasses RL-based optimization on multiple benchmarks without parameter updates.

uate these outputs have become the binding constraint on further progress. Yet most reward models remain scalar black boxes: they compress multi-dimensional human judgments—perceptual quality, instruction faithfulness, physical plausibility, text rendering—into a single unexplained number (Xu et al., 2023; Wu et al., 2025e; Liu et al., 2025b; Wei et al., 2024; Hu et al., 2025). This discards the structured reasoning underlying human preference, leaving generators to exploit shortcut correlations rather than learn principled evaluation criteria (Li et al., 2025). This paper asks: can reward models be made to *reason*—and can their structured critiques not only evaluate but actively *improve* visual generation?

We introduce `RationalRewards`, a reasoning-based reward model that generates structured, multi-dimensional critiques before deriving scores. We argue that this shift from scalar outputs to structured reasoning transforms the reward model from a passive evaluator into a **versatile optimization interface** for visual generation. By producing explicit reasoning, `RationalRewards` unlocks optimization in two complementary spaces:

- **Parameter Space:** Multi-dimensional structured rationales provide semantically grounded, dense feedback for reinforcement learning—replacing opaque scalar gradients prone to reward hacking (Fig. 3), with explanations of *what* to improve and *why*.
- **Prompt Space:** Beyond serving as a reward signal, `RationalRewards` functions as a *post-generation prompt optimizer*. It critiques a generated image, identifies concrete deficiencies, and translates them into targeted prompt revisions in a **Generate–Critique–Refine** loop. Unlike prompt enhancers that rewrite inputs blindly before synthesis (Wang et al., 2025g), this approach is post-hoc and reactive, trading test-time compute for improved fidelity without parameter updates (Snell et al., 2024; Wang et al., 2025c).

Realizing this vision requires a reward model that produces high-quality structured rationales (Mahan et al., 2024; Guo et al., 2025; Zelikman et al., 2022; Wang et al., 2025d), yet human rationale annotations are prohibitively expensive at scale. We observe, however, that pairwise preference data is widely available from online AIGC platforms. Leveraging this, we propose **Preference-Anchored Rationalization (PARROT)**, a variational training framework that treats rationales as latent variables and derives an evidence lower bound (ELBO) on observed preferences. The terms of this ELBO map directly onto a simple, scalable pipeline: (1) a teacher VLM generates candidate rationales anchored to known preference labels, (2) a consistency filter rejects hallucinations and retains rationales that are genuinely predictive, and (3) a student model is trained to produce rationales without seeing the answer. This tight theory–practice correspondence (Fig. 4) converts existing preference datasets into high-quality reasoning supervision using 10–20$\times$ less data than comparable scalar reward baselines.

Figure 3: RL (LoRA) training on Qwen-Image using scalar rewards encounter reward hacking (bottom row): as training reward continues to grow, generation quality starts to degenerate, because black box rewards mislead visual generators with biases. In contrast, `RationalRewards` (top row) sustains generation quality with stable reward growth. See Fig. 10 and 11 for more details.

Key results. Instantiated via PARROT on Qwen3-VL-Instruct-8B backbone, `RationalRewards` achieves state-of-the-art preference prediction among open-source reward models, competitive with Gemini-2.5-Pro (Table 1). As an RL reward, it consistently improves generators beyond scalar baselines across both text-to-image and image editing tasks (Tables 2–3). Most interestingly, `RationalRewards`'s Generate–Critique–Refine loop—requiring no parameter updates—matches or exceeds RL-based fine-tuning on several benchmarks, suggesting that structured critiques can unlock latent generator capabilities that suboptimal prompts fail to elicit. Additional uses, including data filtering and regional critique grounding, are illustrated in Appendix H.

2 Method

We introduce **Preference-Anchored Rationalization (PARROT)**, a framework that trains reward models to produce explicit, multi-dimensional rationales before scores (Zelikman et al., 2022; Wang et al., 2025d). Assessment dimensions—text faithfulness, physical/visual quality, text rendering, and (for editing) image faithfulness—follow the taxonomy of Hu et al. (2025), chosen for coverage of the primary failure modes in current generators.

Since ground-truth rationales are prohibitively expensive to annotate at scale, we formulate rationales as *latent variables* inferred from pairwise preference data via a variational objective. The resulting ELBO (Eq. 1) decomposes into three terms, each corresponding to a concrete pipeline phase (Fig. 4): (1) generate rationales anchored to known preferences, (2) filter for predictive consistency, and (3) distill into a student model. Readers primarily interested in the practical pipeline may consult Fig. 4 and return to the derivation for justification.

Why reasoning rewards resist reward hacking. Scalar rewards expose a single optimization target that can rise when a generator exploits model biases without improving visual quality. `RationalRewards` instead emits evidence-bearing rationales and separate scores for four dimensions before producing an overall assessment. A high score is therefore accompanied by claims about instruction fidelity, preservation, physical quality, or text rendering that can be checked against the image. This structure does not guarantee immunity to reward hacking, but it distributes optimization pressure across interpretable criteria and makes failures easier to diagnose. Empirically, scalar-reward optimization eventually degrades samples even as its reward grows, whereas `RationalRewards`-guided training preserves correspondence between reward and visual quality (Fig. 3); the held-out and benchmark curves show the same alignment in Appendix A.

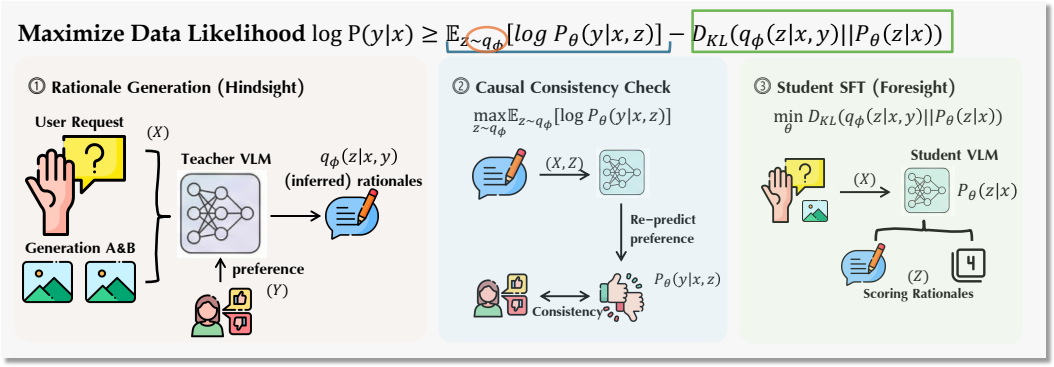


Figure 4: **We implement Preference-Anchored Rationalization as a practical three-phase pipeline.**

Figure 5: Example pointwise scores rated by `RationalRewards` for image/text-to-image generations (rationales omitted). `RationalRewards` evaluates each result across multiple dimensions.

2.1 Variational Framework: The Hindsight-Foresight Decomposition

Let $x = (I_A, I_B, c)$ denote a comparison tuple comprising two generated images and a conditioning user request c (which includes text instructions and, for editing tasks, a source image). Let $y \in \{A \succ B, B \succ A\}$ denote the ground-truth human preference. Unlike reward models that model $P(y|x)$ directly, PARROT introduces a latent natural language rationale z that *explains* preference y.

We treat the rationale z as the explanatory mechanism underlying the preference. We use "explanatory" in the sense of predictive sufficiency: z is a valid rationale if it contains sufficient information to predict preference y from evaluation task x. Our goal is to learn a evaluator reward model (the *Student*) $P_\theta(z, y|x)$ capable of generating the rationale z and predicting the preference y for downstream tasks. To learn this from preference data alone, we maximize the Evidence Lower Bound (ELBO):

$$\mathcal{L}_{\text{ELBO}} = \underbrace{\mathbb{E}_{z\sim q_\phi}[\log P_\theta(y|x,z)]}_{\text{Term 1: Prediction}} - \underbrace{D_{\text{KL}}\big(q_\phi(z|x,y) \,\|\, P_\theta(z|x)\big)}_{\text{Term 2: Regularization}} \tag{1}$$

This derivation reveals a natural "Teacher-Student" structure, decomposing the learning process into two complementary modes:

- **Hindsight (Posterior $q_\phi(z|x,y)$):** Inferring the rationale z when the ground-truth preference y is *known*—analogous to how human experts articulate evidence after forming an initial judgment.
- **Foresight (Prior $P_\theta(z,y|x)$):** Predicting both rationale z and preference y from the input x alone—our target rationalized reward model.

Phase 1: Rationale Generation (Constructing $q_\phi(z|x,y)$). A naive approach prompts a teacher VLM to compare images without guidance, sampling from the prior $p(z|x)$. This is suboptimal: even strong VLMs frequently misjudge subtle visual details (e.g., Table 1 shows even Gemini-3-Pro has 30% disagreement with human preferences). Instead, we use **preference anchoring**: the Teacher (Qwen3-VL-32B-Instruct) generates rationales *conditioned on* the known preference label y, collapsing generation from open-ended evaluation to focused justification. This concentrates probability mass on rationales consistent with the

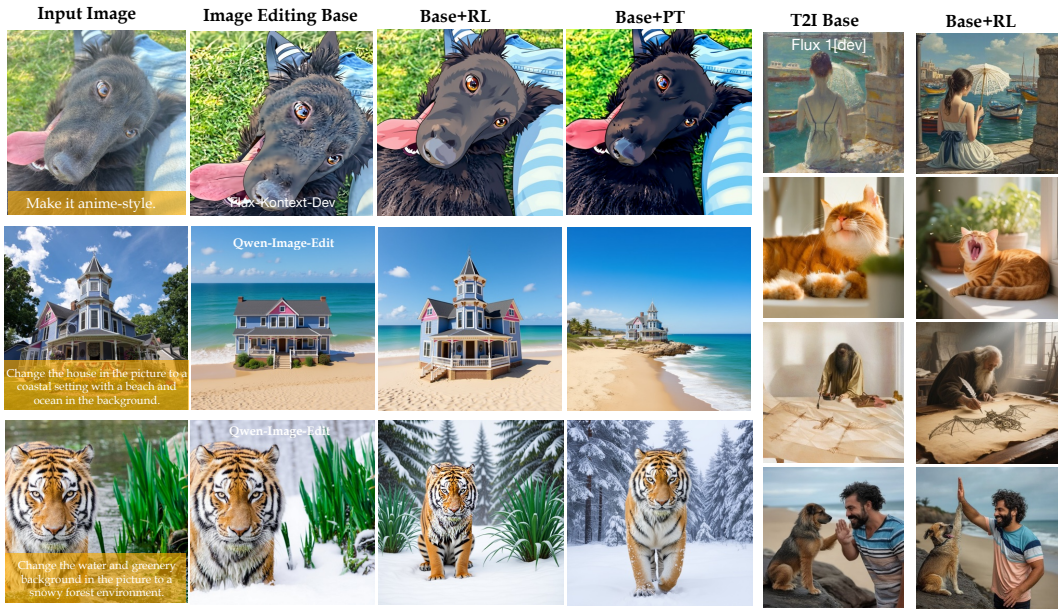


Figure 6: Qualitative results on image/text-to-image tasks optimized with reinforcement learning (RL) and prompt tuning (PT) using `RationalRewards`.

observed label, yielding higher-quality posterior samples than unconditioned generation—confirmed empirically in Table 1.

Prompt structure. The teacher scores Text Faithfulness, Image Faithfulness, Physical and Visual Quality, and Text Rendering on a 1–4 scale, justifies each score, and receives the human preference as an anchor. Complete templates and rubrics are provided in Appendix C.1.

Phase 2. Predictive Consistency Filtering: Maximizing Term 1, $\mathbb{E}_{z\sim q_\phi}[\log P(y|x,z)]$.

While Phase 1 produces rationales that are linguistically plausible, plausibility does not guarantee predictive sufficiency. A rationale z contributes to the ELBO only if it successfully explains y; otherwise, $\log P(y|x,z)$ is low and the corresponding sample degrades the bound. For instance, a VLM might generate a rationale that sounds correct in isolation (e.g., "Image B has distorted text") but does not align with the visual content, or it may ignore the provided preference label altogether.

To enforce predictive sufficiency and thereby maximize Term 1, We enforce that rationales actually explain the preference via a consensus check: the Teacher is re-queried with z *without* the preference label, verifying that z alone suffices to recover y:

$$\mathcal{C}(x,y,z) = \mathbb{I}\left[\arg\max_{y'} P_{\text{Teacher}}(y'|x,z) = y\right] \tag{2}$$

We retain (x,y,z) only if $\mathcal{C}=1$, yielding filtered dataset $\mathcal{D}_{\text{pair}}$. This approximates maximizing $\mathbb{E}_q[\log P(y|x,z)]$ by restricting q_ϕ's support to the high-likelihood region, discarding hallucinated or insufficiently informative rationales.

Phase 3. Foresight Learning: Minimizing Term 2 $D_{\text{KL}}(q_\phi(z|x,y) \,\|\, P_\theta(z|x))$ We train the Student $P_\theta(z|x)$ to generate rationales *without* the preference label via SFT on filtered posterior samples. Since q_ϕ is fixed, minimizing the KL reduces to maximizing $\mathbb{E}_{q_\phi}[\log P_\theta(z|x)]$—precisely the standard SFT objective on filtered samples.

Bridging Pairwise Training and Pointwise Deployment. While we derive the ELBO from pairwise data (which is easier to collect), downstream applications require *pointwise* feedback, e.g., scalar scores for RL training, critiques on individual images for test-time prompt refinement, visual grounding for diagnostic and dense visual rewards. A model

trained solely on pairwise comparisons often fails to critique a single image in isolation, as it overfits to the presence of a contrastive candidate.

We address this with a **Pointwise Projection Strategy**, based on the assumption that pairwise and pointwise assessment share common evaluation principles. We prompt the Teacher to assess each image in isolation, providing the validated pairwise rationale z_{pair} as a reference hint to guide attention toward identified defects. The Teacher articulates absolute scores on a 1–4 scale (with float granularity) across four dimensions: Text Faithfulness, Image Faithfulness, Physical Quality, and Text Rendering. Detailed rubrics are in the appendix. This projection extends beyond the strict pairwise ELBO, but the projected rationales inherit their quality from the ELBO-filtered pairwise rationales and maintain the same predictive relationship between reasoning and scores.

This induces a pointwise dataset $\mathcal{D}_{\text{point}}$. We train the Student jointly on both datasets to enable both pointwise and pairwise assessments: $\mathcal{L}_{\text{SFT}} = \mathbb{E}_{(x,y,z)\sim\mathcal{D}_{\text{point}}\cup\mathcal{D}_{\text{pair}}}\left[-\log P_\theta(z, y|x)\right]$

Why train a reward model rather than use a generic VLM judge. Generic VLMs can produce fluent critiques, but they are not explicitly calibrated to turn small quality differences into stable pointwise scores (Wang et al., 2026b). Repeated judgments of similarly strong images can therefore yield noisy rewards, which is particularly harmful when RL amplifies small score differences. PARROT trains against human preferences and retains only rationales sufficient to recover those preferences; the resulting student learns a consistent mapping from visible evidence to dimension-level scores. This combines the explanatory interface of a VLM with preference calibration and a smaller deployment footprint. The benefit is not explained by model scale alone: although `RationalRewards` is 4× smaller, it is a stronger RL reward than the direct Qwen3-VL-32B judge for every tested generator in Tables 2 and 3, while its reward variance decays during training (Appendix A).

2.2 From Evaluator to Optimizer: Tuning in Parameter Space and Prompt Space

The rationalized reward model enables optimization in two complementary spaces, each suited to various deployment scenarios.

- **Parameter Space (SFT/RL Fine-Tuning).** Multi-dimensional scores provide semantically decomposed reward signals for reinforcement learning, enabling fine-grained feedback across quality dimensions rather than optimization against a single opaque scalar. The structured rationales further serve as natural-language explanations for reward assignments, aiding interpretability and reducing reward hacking (see Fig. 3).
- **Prompt Space (Test-Time Refinement).** Natural-language rationales identify concrete deficiencies in generated images, which we leverage to construct a **Generate–Critique–Refine** loop (Fig. 7): `RationalRewards` critiques an initial generation, and its critique is used to produce a targeted prompt revision for re-generation. This performs $t^* = \arg\max_t R(G(t))$ guided by language rather than numerical gradients, trading test-time compute for quality without parameter updates (Snell et al., 2024). We note that this post-hoc prompt refinement dataset also enables distillation for pre-hoc prompt enhancement models.

This dual-space formulation connects to test-time compute scaling (Snell et al., 2024; Wang et al., 2026a): prompt-space optimization offers an axis for improving generation quality orthogonal to parameter-space training and applicable to any frozen generator. We hypothesize that it is particularly effective when the generator possesses latent capabilities under-elicited by suboptimal prompts—a working hypothesis we examine empirically in Section 3.

Why reasoning rewards enable test-time scaling. Pre-generation prompt rewriters operate without observing the realized failure (Wang et al., 2025g). In contrast, `RationalRewards` conditions its revision on both the request and the generated image, identifies which preference-relevant dimensions fall short, and proposes a targeted correction. The loop therefore spends additional inference compute only after seeing concrete evidence, rather

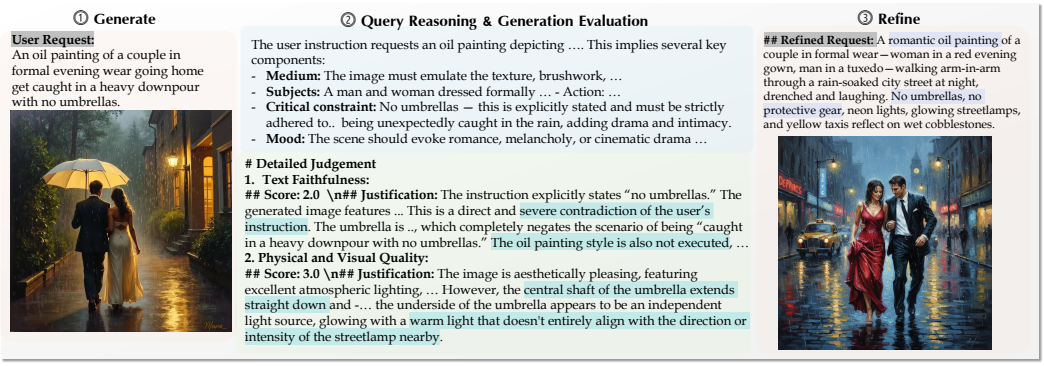


Figure 7: Test-Time Prompt Refinement via "Generate-Critique-Refine" loop with `RationalRewards`.

than expanding every prompt indiscriminately. Preference training also supplies an explicit objective for the rewrite: improve the aspects that predict human choice while retaining satisfied constraints. This distinction motivates why one Generate–Critique–Refine step can compete with weight updates when the generator already has the requisite capability. It remains an empirical mechanism rather than proof of latent capability; Section 3.2 tests the prediction against both pre-generation rewriting and RL.

3 Experiments

Training Data. We use 30K EditReward preference pairs (Wu et al., 2025e) for image editing and 50K HPDv3/RapidData pairs (Ma et al., 2025) for text-to-image generation. These sources contain binary or ranked preferences but no explanations. With Qwen3-VL-32B-Instruct as the teacher, PARROT converts each label-only comparison into preference-anchored reasoning supervision and then removes rationales that cannot predict the label once the anchor is hidden. Approximately 72% survive this consistency check, so filtering discards a meaningful fraction rather than merely accepting teacher outputs. The resulting raw-data scale is 15% of EditReward's 200K editing pairs and below 5% of UnifiedReward's one million generation pairs (Wang et al., 2025h). Some efficiency can come from knowledge already present in the teacher; the equal-data direct-distillation ablation in § 3.1 separates that effect from structured rationalization. Dataset statistics, hyperparameters, hardware, and RL details are in Appendices F and E. **Code, data, and models are available at 🌍 Project Page.**

3.1 Accuracy in Preference Modeling

We first evaluate whether `RationalRewards` produces human-aligned preference judgments. We report pairwise comparison accuracy on three established benchmarks: Multimodal Reward Bench 2 (Hu et al., 2025) and GenAI-Bench (Jiang et al., 2024) and EditReward Bench (Wu et al., 2025e) for both text- and image-to-image generation.

Main Results. As shown in Table 1, our 8B-parameter `RationalRewards` surpasses all open-source scalar reward models by a substantial margin across all three benchmarks, without requiring complex loss designs to handle label noise or annotation ambiguities. Notably, `RationalRewards` outperforms commercial models including Gemini-2.5-Flash and approaches the performance of GPT-5/Gemini-2.5-Pro on preference prediction, offering a cost-effective alternative for quality assessment and evaluation in visual generation.

Ablation of PARROT versus Direct Distillation. To isolate the contribution of PARROT from generic knowledge distillation, we include a baseline that performs direct SFT distillation from Qwen3-VL-32B-Instruct to the same 8B backbone, using the same data volume but without preference-anchored rationalization (marked "Qwen3-VL-32B-Instruct Distillation" in Table 1). This baseline underperforms `RationalRewards` on all benchmarks—by 6.8 points on MMRB2 (T2I) and 17.3 points on GenAI (Edit)—confirming that the structured

Table 1: Comparison of reward models as evaluators. We include Multimodal Reward Bench 2 (MMRB2), EditReward-Bench, and GenAI-Bench. T2I and Edit means text-to-image and image-to-image respectively.

Judge	MMRB2		EditReward	GenAI-Bench	
	T2I	Edit		T2I	Edit
Qwen2.5-VL-7B (Bai et al., 2025)	50.4	57.1	52.69	–	40.48
Qwen2.5-VL-72B	59.1	64.6	63.9	66.6	74.3
Qwen3-VL-8B (Yang et al., 2025)	59.4	61.7	51.9	55.1	50.1
Qwen3-VL-32B	64.1	67.3	64.2	66.9	76.3
EditReward-7B (Wu et al., 2025e)	–	67.2	56.99	–	65.72
UnifiedReward-7B (Wang et al., 2025h)	59.8	–	–	67.9	–
`RationalRewards` (Qwen2.5-VL-7B)	62.3	68.5	63.6	66.4	75.7
`RationalRewards` (Qwen3-VL-8B)	**64.2**	**70.3**	**66.2**	**69.8**	**80.1**
Qwen3-VL-32B-Instruct Distillation	57.4	65.6	56.8	59.3	62.8
Commercial Models					
GPT-4.1	65.8	68.2	58.3	60.5	69.3
Gemini 2.5 Flash (Comanici et al., 2025)	63.1	66.5	58.6	65.8	73.0
Gemini 2.5 Pro	70.5	71.3	71.3	66.2	78.9
Gemini 3 Pro (DeepMind, 2025)	74.4	74.9	72.2	73.1	80.5

rationalization process, not simply access to a larger teacher, drives the performance gains. We also replace the backbone with Qwen2.5-VL-7B-Instruct; the results still exceed prior scalar reward models, clarifying that improvements are attributable to PARROT rather than the specific choice of backbone.

3.2 Optimization in Dual Spaces

Given the strong discriminative performance of `RationalRewards`, we now investigate its utility for improving downstream generation. We explore two complementary optimization strategies: *parameter-space* tuning via RL and *prompt-space* tuning via test-time critique-and-refinement. We evaluate on ImgEdit-Bench (Ye et al., 2025a) and GEdit-Bench-EN (Liu et al., 2025c) for image editing, the UniGen benchmark for text-to-image generation. We also include in the appendix a physics-centric PICA-Bench (Pu et al., 2025) for out-of-distribution stress testing, following each benchmark's prescribed evaluation protocol.

Parameter Space Tuning (RL). We experiment with the recent Diffusion RL approach, DiffusionNFT (Zheng et al., 2025), which samples a group of generations for the same user prompt and optimizes with a weighted diffusion loss. For reproducibility, we include the algorithm and implementation details in the appendix. We use `RationalRewards` to provide dense, per-dimension reward signals for RL fine-tuning and systematically compare against alternative reward models spanning two axes: *scalar vs. reasoning-based* and *generic vs. preference-trained*:

1. **Scalar reward models**: EditReward (Wu et al., 2025e) for image editing and MultiReward (used by DiffusionNFT (Zheng et al., 2025)) for text-to-image generation. These output a single scalar score without natural language reasoning.
2. **Generic reasoning model**: Qwen3-VL-32B-Instruct used directly as a judge. This model can produce natural language critiques but has *not* been trained on preference data via PARROT, isolating the contribution of our training pipeline from raw model scale.

As shown in Tables 3 and 2, RL with `RationalRewards` yields consistent improvements over both base models across nearly all subcategories, surpassing both scalar reward baselines and the generic reasoning baseline. For image editing, `RationalRewards`-guided RL improves Flux.1 Kontext from 3.52 to 3.84 overall on ImgEdit-Bench, outperforming EditReward-guided RL (3.66) by a clear margin. For text-to-image generation,

Table 2: Ablation of RationalRewards for Text-to-image RL on UniGenBench++. We compare scalar reward model *MultiReward* and generic reasoning reward *Qwen3-VL-32B*.

Model	Action	Attribute	Compound	Layout	Grammar	Logic	Relation	Style	Text	World Know.	Overall
FLUX.1-dev	62.24	67.20	45.75	70.84	62.30	29.77	66.88	85.00	32.18	87.50	60.97
+MultiReward	59.78	68.23	44.21	74.37	59.33	28.25	68.35	76.05	36.21	86.03	60.12
+Qwen3-VL-32B	65.47	72.68	53.28	71.82	60.78	33.24	71.85	85.53	42.15	89.47	66.53
+RationalRewards	67.40	76.36	57.67	72.15	60.29	40.53	74.59	87.20	52.57	90.61	**70.34**
SD-3.5-Medium	60.41	66.99	53.35	70.31	59.89	37.73	68.78	89.80	15.23	84.34	60.71
+RationalRewards	64.36	81.49	67.98	75.88	58.68	42.37	75.60	89.60	10.05	91.77	**70.56**
+MultiReward	57.03	66.67	51.03	75.37	57.22	34.86	67.51	77.60	21.84	86.71	62.55
+Qwen3-VL-32B	61.23	74.48	63.85	75.34	59.67	31.23	72.84	84.73	14.87	88.86	66.71
Qwen-Image	82.49	87.93	72.94	86.56	60.96	51.59	80.08	94.70	72.13	94.15	78.36
+MultiReward	79.52	86.45	70.91	88.53	58.43	48.62	80.55	83.75	67.18	92.17	75.61
+Qwen3-VL-32B	81.95	87.45	76.42	87.73	62.93	51.14	81.55	95.20	75.67	95.63	80.17
+RationalRewards	82.11	87.82	78.82	88.07	66.21	52.88	82.21	96.60	79.76	96.57	**82.60**

Table 3: Ablation of RationalRewards as dual-space optimizer on editing tasks. For prompt space tuning, we compare pre-generation PromptEnhance (Wang et al., 2025g). For parameter space tuning, we compare SFT and RL with different rewards. We include OOD physics-aware editing, PICA-Bench with representative aspects (Left), and generic editing benchmarks (Right).

Model	Representative Aspects			Overall
	Light	Reflec.	Deform.	
Flux.1 Kontext [dev]	53.64	43.84	33.74	41.07
+PromptEnhance	55.53	45.87	38.14	45.28
+PT (RationalRewards)	56.87	51.43	41.08	**48.12**
+PICA SFT	51.21	47.22	33.99	41.93
+RL (RationalRewards)	51.75	54.81	39.36	44.25
Qwen-Image-Edit	52.02	49.07	38.14	49.71
+PromptEnhance	58.49	50.42	42.30	50.97
+PT (RationalRewards)	63.34	61.55	43.28	**55.65**
+PICA SFT	60.47	55.19	40.99	52.06
+RL (RationalRewards)	63.07	60.71	41.32	54.11

Model	ImgEdit	GEdit-Bench-EN		
	Overall	G_SC	G_PQ	G_O
Flux.1 Kontext [dev]	3.52	7.16	7.37	6.51
+RL (EditReward)	3.66	7.38	7.53	6.88
+RL (Qwen3-VL-32B)	3.67	7.42	7.48	6.82
+RL (RationalRewards)	3.84	7.75	8.24	**7.37**
+PT (RationalRewards)	**4.01**	7.77	7.61	7.23
Qwen-Image-Edit	4.27	8.00	7.86	7.56
+RL (EditReward)	4.25	8.36	7.91	7.77
+RL (Qwen3-VL-32B)	4.25	8.42	7.83	7.79
+RL (RationalRewards)	4.38	8.74	8.43	8.29
+PT (RationalRewards)	**4.43**	8.94	8.20	**8.33**

RationalRewards lifts FLUX.1-dev from 60.97 to 70.34 on UniGen (+9.37 points), substantially exceeding both MultiReward (60.12) and the direct Qwen3-VL-32B judge (66.53). Notably, the 8B RationalRewards outperforms Qwen3-VL-32B used as a direct judge, confirming that PARROT's structured preference training provides value beyond raw model capacity.

Test-Time Prompt Space Tuning. We leverage the generative nature of RationalRewards in a *Generate–Critique–Refine* protocol: the generator produces an initial image; RationalRewards evaluates it across four dimensions with natural language critique and refinement suggestions; if any dimension score falls below a threshold of 3.0, the refined request is fed back to the generator. This single-iteration loop adds approximately 0.4 seconds of VLM inference overhead per image (via vLLM prefix caching and paged attention), compared to $\sim$384 GPU-hours for RL fine-tuning of a single base model.

Prompt Tuning Matches or Exceeds RL. A striking finding emerges from Table 3: inference-time prompt tuning frequently yields improvements comparable to or exceeding computationally expensive RL. On ImgEdit-Bench, prompt tuning boosts the RL-tuned Flux model from 3.84 to 4.01 overall. For Qwen-Image-Edit, prompt tuning applied on top of RL yields the best overall score of 4.43, with the two methods proving complementary. On GEdit-Bench-EN Overall, prompt tuning (8.33) slightly exceeds RL alone (8.29).

The RL performance ceiling is partly structural: LoRA-based fine-tuning constrains parameter update capacity, and the RL query distribution may not fully cover the evaluation distribution. In contrast, prompt tuning performs per-instance optimization without risk of catastrophic forgetting. More fundamentally, these results suggest a *latent capability hypothesis*: generators already possess the capacity for high-quality outputs, but this capacity is under-elicited by suboptimal prompts. RationalRewards's critique bridges user intent and model capability without weight modification. We note this remains a hypothesis requiring representation-level validation.

4 Related Work

Reward Models for Visual Generation. The standard paradigm in visual generation relies heavily on scalar reward models trained on large-scale human preference datasets. Models such as ImageReward (Xu et al., 2023), VideoReward (Liu et al., 2025b), PickScore (Kirstain et al., 2023), UnifiedReward (Wang et al., 2025h), and EditReward (Wu et al., 2025e) typically function as opaque discriminators, mapping pixel inputs directly to a scalar score. Our work provides an alternative path for reward modeling, shifting the paradigm from scalar regression to rationalization (Zelikman et al., 2022). Generative reward models have also been studied in verifiable domains (Mahan et al., 2024; Guo et al., 2025; Chen et al., 2026).

Training and Test-Time Scaling in Visual Generation. Recent efforts, such as Flow-GRPO (Liu et al., 2025a), DanceGRPO (Xue et al., 2025), Blip3o-Next (Chen et al., 2025), and DiffusionNFT (Zheng et al., 2025; Li et al., 2025), have successfully integrated RL into visual generation, demonstrating significant gains in compositional reasoning and text rendering. While effective, RL is bottlenecked by the quality of the reward model, often suffering from reward hacking when the proxy reward diverges from human preference. Recent works have pivoted toward trading test-time compute for enhanced generation quality. Reflection-Flow (Zhuo et al., 2025) and PromptEnhancer (Wang et al., 2025g) use Chain-of-Thought (CoT) rewriters to expand user prompts into detailed specifications prior to generation. For image editing, Reason-Edit (Yin et al., 2025) introduces a thinking–editing–reflection loop. Most recently, several approaches have begun leveraging the multimodal CoT capabilities of Unified Multimodal Models to iteratively improve visual synthesis at test time (Qin et al., 2025; Wu et al., 2025d; Deng et al., 2025b; Jiang et al., 2025; Ye et al., 2025b; Li et al., 2025). Our work highlights the importance of preference calibration and rationalization in reward models and provides a preference-aware mechanism for converting test-time compute into better generation.

Scope and limitations. The conclusions are bounded in several ways. First, PARROT distills Qwen3-VL-32B-Instruct, so systematic teacher blind spots can propagate to the student even after consistency filtering. Second, the preference datasets encode annotator and source-model biases; we have not completed a comprehensive demographic or cultural bias audit. Third, all reported benchmarks use English instructions and predominantly photorealistic image domains, leaving multilingual transfer and applications to video, 3D, or scientific visualization untested. Fourth, the Generate–Critique–Refine results use a fixed score threshold of 3.0 and one refinement step; threshold sensitivity and multi-step drift require further study. Finally, matching or exceeding RL is consistent with the latent-capability hypothesis but does not establish it causally: extra prompt context is an alternative explanation. Appendix G expands these limitations and discusses broader impact and responsible deployment.

5 Conclusions

We introduced `RationalRewards`, a reasoning-based reward model, and PARROT, a variational pipeline that recovers structured rationale supervision from preference-only data. The approach yields three findings. First, an 8B `RationalRewards` reaches preference accuracy competitive with strong proprietary judges while using 10–20$\times$ less raw training data than comparable scalar-reward baselines. Second, its dimension-specific scores provide more effective RL feedback than scalar rewards and a larger generic VLM judge across text-to-image and image-editing generators. Third, one Generate–Critique–Refine step matches or exceeds RL on several benchmarks without parameter updates and can complement an RL-tuned model. Together, these results position structured rationales as a shared interface between evaluation and optimization: the same output supplies an auditable judgment, decomposed training rewards, and actionable test-time feedback. Establishing how broadly this interface transfers beyond the evaluated generators, languages, and image domains is an important next step.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Jiuhai Chen, Le Xue, Zhiyang Xu, Xichen Pan, Shusheng Yang, Can Qin, An Yan, Honglu Zhou, Zeyuan Chen, Lifu Huang, et al. Blip3o-next: Next frontier of native image generation. *arXiv preprint arXiv:2510.15857*, 2025.

Xiusi Chen, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru Wang, Yu Zhang, Denghui Zhang, Tong Zhang, Hanghang Tong, and Heng Ji. Rm-r1: Reward modeling as reasoning, 2026. URL https://arxiv.org/abs/2505.02387.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Google DeepMind. Google gemini-3 system card. 2025.

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

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*, 2025b.

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.

Google DeepMind. Gemini 2.5 flash image (nano banana). https://ai.google.dev/gemini-api/docs/image-generation, 2025. Google's AI image generation and editing model, officially Gemini 2.5 Flash Image, known by its nickname "Nano Banana". Accessed September 2025.

Jiaxin Guo, Zewen Chi, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. Reward reasoning model, 2025. URL https://arxiv.org/abs/2505.14674.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Yushi Hu, Reyhane Askari-Hemmat, Melissa Hall, Emily Dinan, Luke Zettlemoyer, and Marjan Ghazvininejad. Multimodal rewardbench 2: Evaluating omni reward models for interleaved text and image. *arXiv preprint arXiv:2512.16899*, 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. *arXiv preprint arXiv:2406.04485*, 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.

Dongyang Jin, Ryan Xu, Jianhao Zeng, Rui Lan, Yancheng Bai, Lei Sun, and Xiangxiang Chu. Semantic context matters: Improving conditioning for autoregressive models. *arXiv preprint arXiv:2511.14063*, 2025.

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*, 36:36652–36663, 2023.

Rui Lan, Yancheng Bai, Xu Duan, Mingxing Li, Dongyang Jin, Ryan Xu, Lei Sun, and Xiangxiang Chu. Flux-text: A simple and advanced diffusion transformer baseline for scene text editing. *arXiv preprint arXiv:2505.03329*, 2025.

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.

Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, et al. Uniworld: High-resolution semantic encoders for unified visual understanding and generation. *arXiv preprint arXiv:2506.03147*, 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, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, et al. Improving video generation with human feedback. *arXiv preprint arXiv:2501.13918*, 2025b.

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*, 2025c.

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

Dakota Mahan, Duy Van Phung, Rafael Rafailov, Chase Blagden, Nathan Lile, Louis Castricato, Jan-Philipp Fränken, Chelsea Finn, and Alon Albalak. Generative reward models, 2024. URL https://arxiv.org/abs/2410.12832.

OpenAI. Gpt-image-1. https://platform.openai.com/docs/guides/image-generation?image-generation-model=gpt-image-1, 2025. OpenAI's image generation model. Accessed September 2025.

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.

Yuandong Pu, Le Zhuo, Songhao Han, Jinbo Xing, Kaiwen Zhu, Shuo Cao, Bin Fu, Si Liu, Hongsheng Li, Yu Qiao, et al. Picabench: How far are we from physically realistic image editing? *arXiv preprint arXiv:2510.17681*, 2025.

Luozheng Qin, Jia Gong, Yuqing Sun, Tianjiao Li, Mengping Yang, Xiaomeng Yang, Chao Qu, Zhiyu Tan, and Hao Li. Uni-cot: Towards unified chain-of-thought reasoning across text and vision. *arXiv preprint arXiv:2508.05606*, 2025.

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.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL https://arxiv.org/abs/2408.03314.

Changpeng Wang, Haozhe Wang, Xi Chen, Junhan Liu, Taofeng Xue, Chong Peng, Donglian Qi, Fangzhen Lin, and Yunfeng Yan. From illusion to intention: Visual rationale learning for vision-language reasoning. *arXiv preprint arXiv:2511.23031*, 2025a.

Guo-Hua Wang, Shanshan Zhao, Xinjie Zhang, Liangfu Cao, Pengxin Zhan, Lunhao Duan, Shiyin Lu, Minghao Fu, Jianshan Zhao, Yang Li, and Qing-Guo Chen. Ovis-u1 technical report. *arXiv preprint arXiv:2506.23044*, 2025b.

Haozhe Wang, Jiale Zhou, and Xuming He. Learning context-aware task reasoning for efficient meta-reinforcement learning. *arXiv preprint arXiv:2003.01373*, 2020.

Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *arXiv preprint arXiv:2504.08837*, 2025c.

Haozhe Wang, Haoran Que, Qixin Xu, Minghao Liu, Wangchunshu Zhou, Jiazhan Feng, Wanjun Zhong, Wei Ye, Tong Yang, Wenhao Huang, et al. Reverse-engineered reasoning for open-ended generation. *arXiv preprint arXiv:2509.06160*, 2025d.

Haozhe Wang, Alex Su, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. *arXiv preprint arXiv:2505.15966*, 2025e.

Haozhe Wang, Qixin Xu, Che Liu, Junhong Wu, Fangzhen Lin, and Wenhu Chen. Emergent hierarchical reasoning in llms through reinforcement learning. *arXiv preprint arXiv:2509.03646*, 2025f.

Haozhe Wang, Weijia Feng, Jinpeng Yu, Che Liu, Ping Nie, Fangzhen Lin, Jiaming Liu, Ruihua Huang, Jimmy Lin, Wenhu Chen, et al. Search beyond what can be taught: Evolving the knowledge boundary in agentic visual generation. *arXiv preprint arXiv:2607.05382*, 2026a.

Haozhe Wang, Qixin Xu, Changpeng Wang, Taofeng Xue, Chong Peng, Wenhu Chen, and Fangzhen Lin. Bad seeing or bad thinking? rewarding perception for multimodal reasoning. *arXiv preprint arXiv:2605.14054*, 2026b.

Linqing Wang, Ximing Xing, Yiji Cheng, Zhiyuan Zhao, Donghao Li, Tiankai Hang, Jiale Tao, Qixun Wang, Ruihuang Li, Comi Chen, et al. Promptenhancer: A simple approach to enhance text-to-image models via chain-of-thought prompt rewriting. *arXiv preprint arXiv:2509.04545*, 2025g.

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

Cong Wei, Zheyang Xiong, Weiming Ren, Xinrun Du, Ge Zhang, and Wenhu Chen. Omniedit: Building image editing generalist models through specialist supervision. *arXiv preprint arXiv:2411.07199*, 2024.

Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, et al. Hunyuanvideo 1.5 technical report. *arXiv preprint arXiv:2511.18870*, 2025a.

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, 2025b. 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, Ze Liu, Ziyi Xia, Chaofan Li, Haoge Deng, Jiahao Wang, Kun Luo, Bo Zhang, Defu Lian, Xinlong Wang, Zhongyuan Wang, Tiejun Huang, and Zheng Liu. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025c.

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*, 2025d.

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*, 2025e.

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*, 36:15903–15935, 2023.

Ryan Xu, Dongyang Jin, Yancheng Bai, Rui Lan, Xu Duan, Lei Sun, and Xiangxiang Chu. Scalar: Scale-wise controllable visual autoregressive learning. *arXiv preprint arXiv:2507.19946*, 2025.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 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, 2025a. URL https://arxiv.org/abs/2505.20275.

Zixuan Ye, Quande Liu, Cong Wei, Yuanxing Zhang, Xintao Wang, Pengfei Wan, Kun Gai, and Wenhan Luo. Visual-aware cot: Achieving high-fidelity visual consistency in unified models. *arXiv preprint arXiv:2512.19686*, 2025b.

Fukun Yin, Shiyu Liu, Yucheng Han, Zhibo Wang, Peng Xing, Rui Wang, Wei Cheng, Yingming Wang, Aojie Li, Zixin Yin, et al. Reasonedit: Towards reasoning-enhanced image editing models. *arXiv preprint arXiv:2511.22625*, 2025.

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.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. UltraEdit: Instruction-based fine-grained image editing at scale. *arXiv preprint arXiv:2407.05282*, 2024.

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.

Le Zhuo, Liangbing Zhao, Sayak Paul, Yue Liao, Renrui Zhang, Yi Xin, Peng Gao, Mohamed Elhoseiny, and Hongsheng Li. From reflection to perfection: Scaling inference-time optimization for text-to-image diffusion models via reflection tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15329–15339, 2025.

Table 4: Text-to-image generation results on UniGen benchmark. We report category-level scores and overall performance. **Action** is the average of Hand, Full Body, Animal, Non Contact, Contact, and State. **Layout** is the average of 2D and 3D.

Model	Action	Attribute	Compound	Layout	Grammar	Logic	Relation	Style	Text	World Know.	Overall
Open-source / Commercial Models											
Nano Banana Pro (Google DeepMind, 2025)	91.30	91.95	92.91	93.30	89.59	80.24	95.43	99.30	95.65	97.47	92.72
Seedream-4-5-251128	88.06	91.03	90.08	92.55	84.09	73.17	90.61	99.20	91.67	96.35	89.70
UniWorld-V1 (Lin et al., 2025)	66.94	70.62	54.51	68.96	63.77	38.41	67.13	91.10	26.44	82.91	63.11
OmniGen2 (Wu et al., 2025c)	62.68	72.12	56.31	71.54	59.89	32.50	68.27	91.90	29.02	86.39	63.09
Bagel (Deng et al., 2025a)	61.89	67.73	56.86	76.59	65.85	23.85	70.64	90.08	0.00	85.42	59.91
BLIP3-o (Chen et al., 2025)	64.25	64.77	54.57	67.23	69.05	36.78	65.99	92.81	0.00	79.97	59.57
Emu3 (Sheynin et al., 2024)	40.21	50.11	36.21	43.87	50.67	19.32	48.60	87.50	1.15	76.42	45.42
SDXL (Podell et al., 2023)	34.44	44.66	26.68	30.70	48.48	10.34	46.37	87.45	0.00	72.28	40.22
Train-Time Scaling w/ RationalRewards											
FLUX.1-dev	62.24	67.20	45.75	70.84	62.30	29.77	66.88	85.00	32.18	87.50	60.97
+MultiReward	59.78	68.23	44.21	74.37	59.33	28.25	68.35	76.05	36.21	86.03	60.12
+Qwen3-VL-32B	65.47	72.68	53.28	71.82	60.78	33.24	71.85	85.53	42.15	89.47	66.53
+RationalRewards	67.40	76.36	57.67	72.15	60.29	40.53	74.59	87.20	52.57	90.61	**70.34**
SD-3.5-Medium	60.41	66.99	53.35	70.31	59.89	37.73	68.78	89.80	15.23	84.34	60.71
+RationalRewards	64.36	81.49	67.98	75.88	58.68	42.37	75.60	89.60	10.05	91.77	**70.56**
+MultiReward	57.03	66.67	51.03	75.37	57.22	34.86	67.51	77.60	21.84	86.71	62.55
+Qwen3-VL-32B	61.23	74.48	63.85	75.34	59.67	31.23	72.84	84.73	14.87	88.86	66.71
Qwen-Image	82.49	87.93	72.94	86.56	60.96	51.59	80.08	94.70	72.13	94.15	78.36
+MultiReward	79.52	86.45	70.91	88.53	58.43	48.62	80.55	83.75	67.18	92.17	75.61
+Qwen3-VL-32B	81.95	87.45	76.42	87.73	62.93	51.14	81.55	95.20	75.67	95.63	80.17
+RationalRewards	82.11	87.82	78.82	88.07	66.21	52.88	82.21	96.60	79.76	96.57	**82.60**

Table 5: We perform RL tuning and test-time prompt tuning to test `RationalRewards` on image editing. On ImgEdit-Bench and GEdit-Bench-EN, trading test-time evaluation for better generation yields surprising gains.

Model	ImgEdit-Bench										GEdit-Bench-EN		
	Add	Adjust	Extract	Replace	Remove	Background	Style	Compose	Action	Overall	G_SC	G_PQ	G_O
AnyEdit (Yu et al., 2025)	3.18	2.95	1.88	2.47	2.23	2.23	2.85	1.56	2.65	2.45	3.18	5.82	3.21
UltraEdit (Zhao et al., 2024)	3.44	2.81	2.13	2.96	1.45	2.86	3.76	1.91	2.98	2.70	-	-	-
Step1X-Edit (Liu et al., 2025c)	3.88	3.14	1.76	3.40	2.41	3.16	4.63	2.64	2.52	3.06	7.66	7.35	6.97
BAGEL (Deng et al., 2025a)	3.56	3.31	1.70	3.30	2.62	3.24	4.49	2.38	4.17	3.20	7.36	6.83	6.52
OmniGen2 (Wu et al., 2025c)	3.57	3.06	1.77	3.74	3.20	3.57	4.81	2.52	4.68	3.44	7.16	6.77	6.41
Ovis-U1 (Wang et al., 2025b)	4.13	3.62	2.98	4.45	4.06	4.22	4.69	3.45	4.61	4.00	-	-	6.42
GPT-Image-1 (OpenAI, 2025)	4.61	4.33	2.90	4.35	3.66	4.57	4.93	3.96	4.89	4.20	7.85	7.62	7.53
Train/Test Time Scaling /w `RationalRewards`													
Flux.1 Kontext [dev]	3.76	3.45	2.15	3.98	2.94	3.78	4.38	2.96	4.26	3.52	7.16	7.37	6.51
+RL (EditReward)	3.91	3.83	2.39	4.15	2.99	3.99	4.56	2.73	4.11	3.66	7.38	7.53	6.88
+RL (Qwen3-VL-32B)	3.95	3.90	2.41	4.12	2.95	3.96	4.45	2.82	4.30	3.67	7.42	7.48	6.82
+RL (`RationalRewards`)	4.21	4.34	2.68	4.33	2.92	4.05	4.37	3.09	4.41	3.84	7.75	8.24	**7.37**
+PT (`RationalRewards`)	3.96	4.16	3.37	4.38	3.84	4.12	4.55	2.70	4.29	**4.01**	7.77	7.61	7.23
Qwen-Image-Edit	4.38	4.16	3.43	4.66	4.14	4.38	4.81	3.18	4.69	4.27	8.00	7.86	7.56
+RL (EditReward)	4.34	4.22	3.87	4.67	4.18	4.20	4.83	3.36	4.54	4.25	8.36	7.91	7.77
+RL (Qwen3-VL-32B)	4.40	4.18	3.35	4.60	4.10	4.35	4.80	3.10	4.72	4.25	8.42	7.83	7.79
+RL (`RationalRewards`)	4.41	4.32	4.09	4.63	4.26	4.25	4.91	3.44	4.52	4.38	8.74	8.43	8.29
+ PT (`RationalRewards`)	4.46	4.40	4.18	4.63	4.27	4.40	4.88	3.27	4.54	**4.43**	8.94	8.20	**8.33**

A Extended Experimental Results

Full Text-to-Image Results on UniGenBench++ Table 4 provides the complete UniGenBench++ results across all categories and model variants.

Full Image Editing Results Table 5 provides the complete results on generic image editing benchmarks.

Full PICA-Bench Results Table 6 provides the complete PICA-Bench results across all physics-aware aspects, extending the representative results shown in Table 3 (left panel) of the main text.

Training Curves and Visualizations This section provides training curves referenced in Section 3.2 of the main text, demonstrating that RationalRewards provides stable reward gradients with reduced reward hacking, as shown in Fig. 8. Qualitative Results throughout RL training are visualized in Fig. 9.

Reward Hacking and Visualizations. Fig. 10 compares `RationalRewards` with representative scalar reward models used in text-to-image and image-to-image generation RL. `RationalRewards` demonstrates nice properties of smooth, converging reward curve and standard-deviation curve. In contrast, EditReward remains high variances, leading to unstable reward curve. MultiReward exhibits low variances because it does not suffice to differantiate generations of high-capability generators. Fig. 11 shows clear visual evidence of reward hacking.

Table 6: We test OOD Generalization of `RationalRewards` on physics-aware editing tasks (PICABench).

Model	LightProp	LightSrcEff	Reflection	Refraction	Deformation	Causality	GlobalStateTrans	LocalStateTrans	Overall
Nano Banana	53.27	54.45	55.99	56.58	47.68	58.93	58.17	52.81	55.40
GPT-Image-1	59.56	61.99	52.61	61.84	44.99	53.16	70.53	51.56	57.83
Nano Banana Pro	59.32	64.69	61.38	60.09	53.55	64.70	72.08	63.41	63.29
Seedream 4.0	58.84	66.04	58.85	62.72	50.12	67.09	77.37	63.62	64.91
GPT-Image-1.5	62.95	71.43	61.21	62.28	57.18	67.23	76.93	66.11	67.01
Uniworld-V1	37.77	34.50	37.44	30.70	30.32	34.18	28.81	38.67	33.80
Bagel	54.48	63.34	55.28	55.70	42.05	52.32	68.43	54.05	56.44
Bagel-Think	42.86	52.29	43.17	48.25	40.10	38.40	53.86	46.99	45.91
OmniGen2	51.09	47.98	48.74	45.18	42.79	48.24	52.76	42.41	48.18
Step1X-Edit	43.10	52.29	47.05	47.37	40.34	46.69	57.95	47.19	48.83
Flux.1 Kontext [dev]	48.43	53.64	43.84	43.86	33.74	34.04	41.06	37.01	41.07
+PromptEnhance	48.91	55.53	45.87	43.86	38.14	44.30	44.15	43.87	45.28
+PT (`RationalRewards`)	53.27	56.87	51.43	40.35	41.08	43.60	55.30	43.04	**48.12**
+PICA SFT	49.64	51.21	47.22	46.49	33.99	35.44	39.29	40.75	41.93
+RL (`RationalRewards`)	50.85	51.75	54.81	41.23	39.36	36.99	43.27	35.76	44.25
Qwen-Image-Edit	52.54	52.02	49.07	57.46	38.14	42.62	57.73	47.82	49.71
+PromptEnhance	54.24	58.49	50.42	49.12	42.30	43.46	57.40	50.31	50.97
+PT (`RationalRewards`)	61.26	63.34	61.55	55.70	43.28	46.27	57.28	56.55	**55.65**
+PICA SFT	52.89	60.47	55.19	56.12	40.99	46.24	55.25	51.27	52.06
+RL (`RationalRewards`)	59.56	63.07	60.71	55.26	41.32	45.85	56.40	49.69	54.11

Figure 8: RL with `RationalRewards` on Qwen-Image (text-to-image generator) and Flux-Kontext [dev] (image-to-image editing). The reward standard-deviation gradually decays as training proceeds. Crucially, the evaluation reward curve on held-out eval-set align well with the score curve on target test benchmarks.

Critique Visualization. We provide additional example use case of `RationalRewards`, which visualizes problematic regions and grounds its scoring in the image. Specifically, `RationalRewards` is further fine-tuned to generate structured referring expressions that describe problematic regions. These expressions are used by GroundingDINO to localize the regions, and the resulting bounding boxes are then used by SAM to produce segmentation masks as show in Figure 12.

B ELBO Derivation and Theoretical Details

This appendix provides the complete derivation of the Evidence Lower Bound (ELBO) presented in Eq. 1 of the main text (Section 2.1) and discusses the theoretical assumptions underlying the pointwise projection strategy.

B.1 Full ELBO Derivation

We begin from the log marginal likelihood of the observed preference y given input $x = (I_A, I_B, c)$, where I_A, I_B are two generated images and c is the conditioning user request. We introduce a latent natural language rationale z that explains the preference:

$$\log P_\theta(y \mid x) = \log \int P_\theta(y, z \mid x)\, dz. \tag{3}$$

Figure 9: The evolution of generation quality of RL using `RationalRewards`

Since this marginal is intractable (the integral is over all possible natural-language rationales), we introduce a variational distribution $q_\phi(z \mid x, y)$—the *posterior* over rationales given both the input and the known preference. Multiplying and dividing inside the integral:

$$\log P_\theta(y \mid x) = \log \int q_\phi(z \mid x, y) \frac{P_\theta(y, z \mid x)}{q_\phi(z \mid x, y)} \, dz = \log \, \mathbb{E}_{z \sim q_\phi(\cdot \mid x, y)} \left[\frac{P_\theta(y, z \mid x)}{q_\phi(z \mid x, y)} \right]. \tag{4}$$

Applying Jensen's inequality ($\log \mathbb{E}[\cdot] \geq \mathbb{E}[\log \cdot]$, since log is concave):

$$\log P_\theta(y \mid x) \geq \mathbb{E}_{z \sim q_\phi} \left[\log \frac{P_\theta(y, z \mid x)}{q_\phi(z \mid x, y)} \right] \equiv \mathcal{L}_{\text{ELBO}}. \tag{5}$$

We now decompose the joint $P_\theta(y, z \mid x)$ using the chain rule $P_\theta(y, z \mid x) = P_\theta(y \mid x, z) \cdot P_\theta(z \mid x)$:

$$\begin{aligned}
\mathcal{L}_{\text{ELBO}} &= \mathbb{E}_{z \sim q_\phi} \left[\log P_\theta(y \mid x, z) + \log P_\theta(z \mid x) - \log q_\phi(z \mid x, y) \right] && (6) \\
&= \underbrace{\mathbb{E}_{z \sim q_\phi} \left[\log P_\theta(y \mid x, z) \right]}_{\text{Term 1: Prediction}} + \mathbb{E}_{z \sim q_\phi} \left[\log \frac{P_\theta(z \mid x)}{q_\phi(z \mid x, y)} \right] && (7) \\
&= \underbrace{\mathbb{E}_{z \sim q_\phi} \left[\log P_\theta(y \mid x, z) \right]}_{\text{Term 1: Prediction}} - \underbrace{D_{\text{KL}} \big(q_\phi(z \mid x, y) \,\|\, P_\theta(z \mid x) \big)}_{\text{Term 2: Regularization}}, && (8)
\end{aligned}$$

which yields Eq. 1 in the main text.

Tightness of the Bound. The gap between the ELBO and the true log-likelihood is given exactly by the KL divergence between the variational posterior and the true posterior:

$$\log P_\theta(y \mid x) = \mathcal{L}_{\text{ELBO}} + D_{\text{KL}} \big(q_\phi(z \mid x, y) \,\|\, P_\theta(z \mid x, y) \big). \tag{9}$$

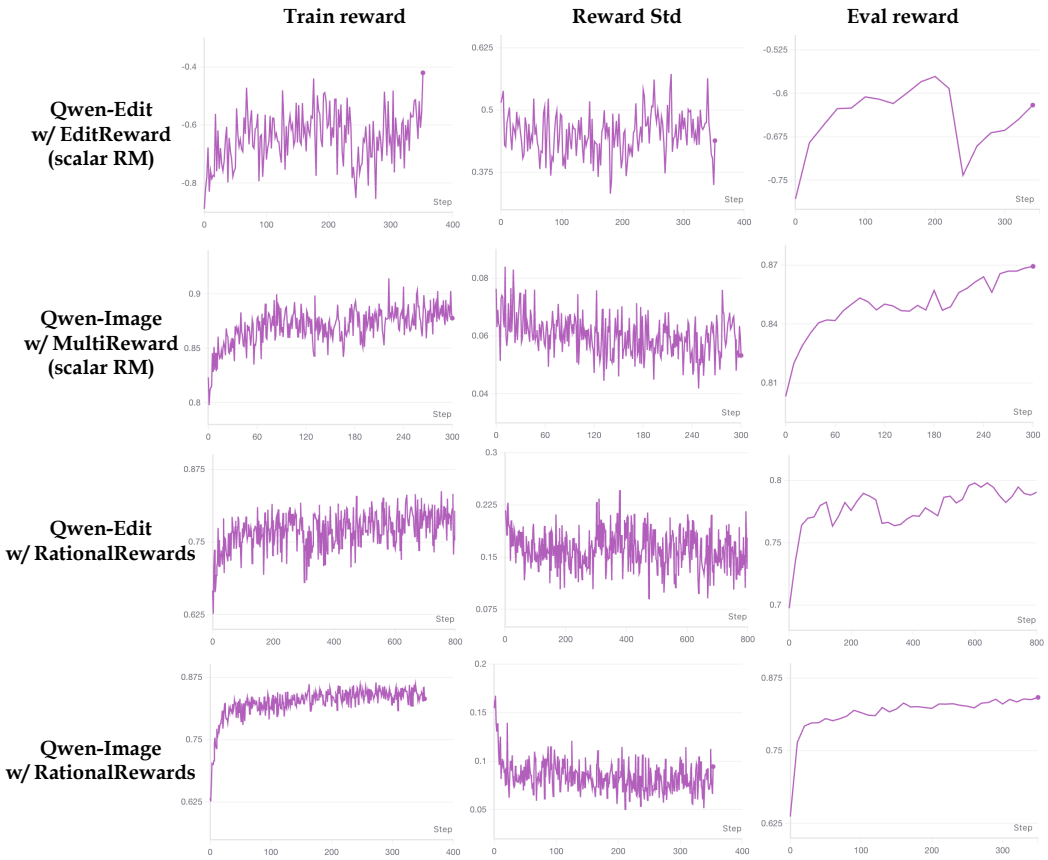


Figure 10: Training curves comparison between `RationalRewards` and scalar reward model, EditReward (Wu et al., 2025e) and MultiReward used in DiffusionNFT Zheng et al. (2025).

This follows directly from the definition of KL divergence:

$$D_{\mathrm{KL}}\big(q_\phi \,\|\, P_\theta(\cdot \mid x, y)\big) = \mathbb{E}_{q_\phi}\left[\log \frac{q_\phi(z \mid x, y)}{P_\theta(z \mid x, y)}\right] \tag{10}$$

$$= \mathbb{E}_{q_\phi}\left[\log q_\phi(z \mid x, y) - \log P_\theta(z, y \mid x) + \log P_\theta(y \mid x)\right] \tag{11}$$

$$= -\mathcal{L}_{\mathrm{ELBO}} + \log P_\theta(y \mid x). \tag{12}$$

Since $D_{\mathrm{KL}} \geq 0$ and $\log P_\theta(y \mid x)$ is fixed with respect to ϕ, maximizing the ELBO is equivalent to minimizing the KL divergence between the variational posterior $q_\phi(z \mid x, y)$ and the true posterior $P_\theta(z \mid x, y)$.

Mapping ELBO Terms to Pipeline Phases. The three terms of the decomposition correspond directly to the three phases of the PARROT pipeline (Figure 3):

1. **Phase 1 (Rationale Generation)** constructs the variational posterior $q_\phi(z \mid x, y)$ by prompting a teacher VLM with preference-anchored instructions. The preference label y is provided as a hint, focusing generation on rationales consistent with the observed preference.
2. **Phase 2 (Consistency Filtering)** maximizes Term 1, $\mathbb{E}_{q_\phi}[\log P_\theta(y \mid x, z)]$, by retaining only rationales z for which the preference y can be recovered from (x, z) alone (Eq. 2). This restricts q_ϕ's effective support to the high-likelihood region, ensuring predictive sufficiency.
3. **Phase 3 (Foresight Distillation)** minimizes Term 2, $D_{\mathrm{KL}}(q_\phi(z \mid x, y) \| P_\theta(z \mid x))$, by training the student model $P_\theta(z \mid x)$ to generate rationales without access to y. Since q_ϕ

Figure 11: Text-to-Image RL using scalar reward model demonstrates reward hacking – while the reward increases, the visual quality of generations degrades notably.

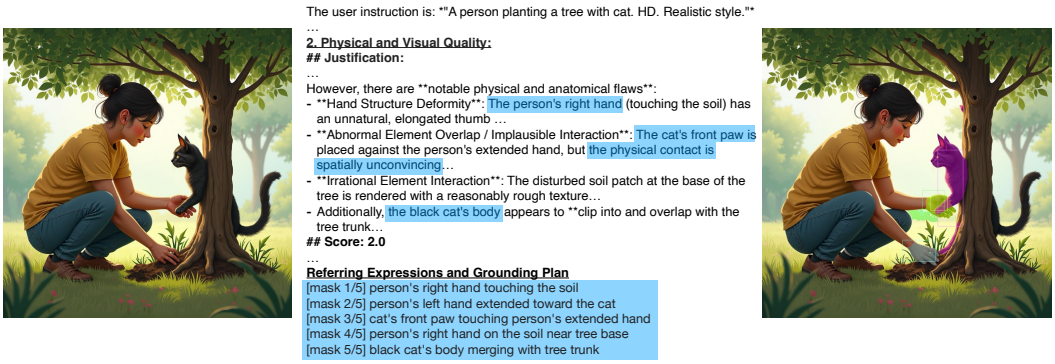


Figure 12: Illustration of Critique Visualization.`RationalRewards` first analyzes the image and provides critique rationales, then summarizes them and generates referring expressions for GroundingDINO and SAM to produce segmentation masks for problematic regions.

is fixed, this reduces to maximizing $\mathbb{E}_{q_\phi}[\log P_\theta(z \mid x)]$, which is precisely the standard supervised fine-tuning (SFT) objective on the filtered posterior samples.

Factorization Assumption. The derivation assumes the joint factorizes as $P_\theta(z, y \mid x) = P_\theta(y \mid x, z) \cdot P_\theta(z \mid x)$, i.e., the model first generates a rationale z given the input x, then predicts the preference y conditioned on both. This autoregressive factorization is natural for language models, where z (the rationale) is generated token-by-token before the preference prediction y. The factorization encodes the causal assumption that the rationale mediates the preference judgment—the model must "show its work" before committing to a decision (Wang et al., 2025a; 2020).

B.2 Justification for Pointwise Projection

The pointwise projection strategy (Section 2.1, main text) extends the pairwise ELBO framework to absolute scoring of individual images. We discuss the assumptions underlying this extension.

Shared Evaluation Principles. The core assumption is that the evaluation criteria underlying pairwise preference (e.g., "Image A has better text faithfulness than Image B because...") are transferable to absolute assessment (e.g., "This image has a text faithfulness score of 3.2 because..."). This is grounded in the observation that the same rubric dimensions—text faithfulness, image faithfulness, physical quality, and text rendering—apply in both settings, differing only in whether the assessment is relative or absolute.

Role of Pairwise Rationales as Reference Hints. During pointwise projection, the validated pairwise rationale z_{pair} serves as a reference hint to guide the teacher's attention toward specific defects or qualities already identified in the pairwise comparison. This anchoring reduces the variance of pointwise assessments by providing concrete evidence (e.g., "as noted in the comparison, the text rendering in this image has minor misspellings") rather than requiring the teacher to identify all issues from scratch. The quality of pointwise rationales thus inherits from the ELBO-filtered pairwise rationales.

Potential Failure Modes. We acknowledge two potential failure modes: (1) *calibration drift*, where the relative ranking between two images is correct but the absolute scores are miscalibrated (e.g., both images receive high scores despite one being clearly inferior); and (2) *context dependence*, where the teacher's absolute assessment is influenced by the identity of the comparison partner in the pairwise rationale, rather than being truly absolute. We mitigate (1) through float-valued scoring with detailed rubric anchors (Appendix D) and (2) by instructing the teacher to assess "as if by your own judgement" independently of the reference hint.

C Prompt Templates

This appendix provides the complete prompt templates used across all phases of the PARROT pipeline and the Generate–Critique–Refine (GCR) loop, referenced in Section 2.1 of the main text.

C.1 Phase 1: Pairwise Rationale Generation Prompt

The following prompt is used to query the teacher VLM (Qwen3-VL-32B-Instruct) for pairwise rationale generation with preference anchoring. The main text (Section 2.1) shows an abbreviated version; below is the complete template.

Pairwise Rationale Generation — Image Editing Variant

```
User Instruction: {instruction}

You are provided with three images:
1. The Source Image (First image)
2. Edited Image A (Second image)
```

3. Edited Image B (Third image)
Your task is to compare two edited images (Edited Image A and Edited Image B) against the Source Image and the User Instruction.

To do this, you must assess each image on four critical aspects, provide justifications and scores in 1--4 scale, and determine which image is better for each aspect.

About the scores: you should try to give **float scores**. For example, float values are important to reflect fine-grained preferences when you compare two edited images.

Critical Aspects & Scoring Rubric

1. Text Faithfulness (How accurately does the output follow the instruction?)
- **4 (Full match):** All key elements (objects, colors, actions) are represented exactly as described. No hallucinations or unrequested changes.
- **3 (Minor mismatch):** Most key elements are present, but minor details are missing, incorrect, or slightly inaccurate.
- **2 (Some mismatch):** Some key elements are missing, altered, or interpreted incorrectly.
- **1 (Major deviations):** Key elements are completely missing, altered, or contradicted. Instruction is ignored.

2. Image Faithfulness (How well are the non-edited parts and key input elements preserved?)
- **4 (Uses input fully):** All relevant elements from the input are accurately preserved or transformed as instructed.
- **3 (Minor mismatch):** Most relevant elements are preserved, but a few aspects are missing or incorrectly handled.
- **2 (Partial mismatch):** Some elements are carried over, but key aspects of the original image are lost or distorted.
- **1 (Fails to use input):** Key elements of the input image are ignored, misinterpreted, or destroyed.

3. Physical and Visual Quality (Technical errors, composition, realism, and physics)
- **4 (No noticeable flaws):** The image is physically plausible. No visible artifacts.
- **3 (Minor flaws):** Small inaccuracies that are noticeable but not strongly disruptive.
- **2 (Some flaws):** Clear physical or visual errors that disrupt the image.
- **1 (Severe flaws):** Major physical/visual errors.

4. Text Rendering (Only if the instruction involves generating text)
- **4 (Full match):** Text is correct, legible, and integrated well.
- **3 (Mostly match):** Minor misspellings or inconsistent capitalization.
- **2 (Partial match):** Major misspellings or distorted text.
- **1 (Major deviations):** Text is unreadable, severely distorted, or missing. (Use N/A if no text generation is required).

Hint: human preference is: {**label**}

Output your evaluation in the following format:
[Understanding of the user request]
 # Detailed Judgement
1. Text Faithfulness:
Justification: [Detailed comparison]
Score A: [float] **## Score B:** [float] **## Winner:** [A/B/Tie]
2--4. (same structure for remaining aspects)
 # Summary: [Overall comparison summary]

Text-to-Image Variant. For text-to-image generation, the prompt is modified as follows: (1) only two images are provided (Generated Image A and Generated Image B) without

a source image; (2) the "Image Faithfulness" dimension is replaced with N/A since there is no source image to preserve; and (3) the task description is adjusted to "compare two generated images against the User Instruction."

C.2 Phase 2: Consistency Check Prompt

The following prompt is used to re-query the teacher VLM *without* the preference label to verify that the generated rationale z alone suffices to recover the preference y (Eq. 2).

Consistency Check Prompt

```
User Instruction: {instruction}

You are provided with the following evaluation of two edited images:

{rationale_z}

Based on the above evaluation, which image is preferred overall?

Answer with ONLY one of the following:
- ``A is preferred''
- ``B is preferred''
- ``Tie''
```

The rationale z is presented in its entirety (including per-dimension justifications, scores, and summary). The teacher must predict the preference from the rationale alone. If the predicted preference matches the ground-truth label y, the sample passes the consistency check ($C = 1$ in Eq. 2).

C.3 Pointwise Projection Prompt

The following prompt is used to obtain pointwise (absolute) assessments from the teacher VLM, guided by the validated pairwise rationale as a reference hint.

Pointwise Projection (Part 1)

```
User Instruction: {instruction}

Note: The reference comment above was based on a comparison between two images.
The edited image you are currently assessing is referred to as ``{image_label}''
in that
comment.  Use this comment as a reference to help you evaluate the Edited Image
more
accurately.

You are provided with two images:
1. The Source Image (First image)
2. The Edited Image (Second image)

Your task is to evaluate the Edited Image against the Source Image and the User
Instruction. To do this, you must first assess the image on four critical aspects,
provide justifications and absolute scores in 1--4 scale.
About the scores: you should try to give float scores. For example, float values
are
important to reflect fine-grained preferences when you compare two edited images.

### Critical Aspects & Scoring Rubric

1. Text Faithfulness (How accurately does the output follow the instruction?)
- 4 (Full match): All key elements (objects, colors, actions) are represented
exactly as described. No hallucinations or unrequested changes.
```

- **3 (Minor mismatch):** Most key elements are present, but minor details are missing, incorrect, or slightly inaccurate.
- **2 (Some mismatch):** Some key elements are missing, altered, or interpreted incorrectly.
- **1 (Major deviations):** Key elements are completely missing, altered, or contradicted. Instruction is ignored.

2. Image Faithfulness (How well are the non-edited parts and key input elements preserved?)
- **4 (Uses input fully):** All relevant elements from the input (background, style, lighting, identity) are accurately preserved or transformed as instructed.
- **3 (Minor mismatch):** Most relevant elements are preserved, but a few aspects (e.g., background details, lighting consistency) are missing or incorrectly handled.
- **2 (Partial mismatch):** Some elements are carried over, but key aspects of the original image are lost or distorted.
- **1 (Fails to use input):** Key elements of the input image are ignored, misinterpreted, or destroyed.

3. Physical and Visual Quality (Technical errors, composition, realism, and physics)
- **4 (No noticeable flaws):** The image is physically plausible (correct lighting, shadows, geometry, anatomy). No visible artifacts (seams, blurring, noise).
- **3 (Minor flaws):** Small inaccuracies that are noticeable but not strongly disruptive (e.g., slight lighting mismatch, minor texture issues).
- **2 (Some flaws):** Clear physical or visual errors that disrupt the image (e.g., incorrect perspective, ‘‘floating’’ objects, wrong shadow direction, obvious seams).
- **1 (Severe flaws):** Major physical/visual errors (e.g., impossible geometry, distorted anatomy, garbled objects, severe artifacts).

Pointwise Projection (Part 2)

4. Text Rendering (Only if the instruction involves generating text)
- **4 (Full match):** Text is correct, legible, and integrated well.
- **3 (Mostly match):** Minor misspellings or inconsistent capitalization.
- **2 (Partial match):** Major misspellings or distorted text.
- **1 (Major deviations):** Text is unreadable, severely distorted, or missing.

(Use N/A if no text generation is required).

Here is a relevant comment. The comment compares the edited image (referred to as ‘‘{image_label}’’) with another edited image:
{reference_comment}
Note: The relevant comment is a hint for you. You can leverage what is useful in it to generate your response. But you MUST NOT mention that this relevant comment is provided when writing the below justifications. Act as if you assess by your own judgement.

Output your evaluation in the following format:
[understanding the user request, and what needs to be considered during image editing]
 # Detailed Judgement
1. Text Faithfulness:
Score: [float score]
Justification: [Detailed explanation of the score]
2. Image Faithfulness:
Score: [float score]
Justification: [Detailed explanation of the score]
3. Physical and Visual Quality:
Score: [float score]
Justification: [Detailed explanation of the score]

```
4. Text Rendering:
## Score: [ float score or N/A ]
## Justification: [Detailed explanation of the score]
# Summary: [Summary of the evaluation]
```

C.4 Generate–Critique–Refine (GCR) Loop Prompts

The GCR loop at test time (Section 2.2, Figure 6) uses the trained RationalRewards model in two stages. First, the *critique prompt* evaluates a single generated image across four dimensions with natural language justification. Then, the model generates a *refinement* including a summary of deficiencies and a revised user prompt.

GCR Critique and Refinement Prompt (Part 1)

```
User Instruction: {instruction}

You are provided with two images:
1. The Source Image (First image)
2. The Edited Image (Second image)
Your task is to evaluate the Edited Image against the Source Image and the User
Instruction.

To do this, you must first assess the image on four critical aspects, provide
justifications and absolute scores in 1--4 scale.

About the scores: you should try to give float scores. For example, float values
are important to reflect fine-grained preferences when you compare two edited
images.

### Critical Aspects & Scoring Rubric

1. Text Faithfulness (How accurately does the output follow the instruction?)
- 4 (Full match): All key elements (objects, colors, actions) are represented
exactly as described. No hallucinations or unrequested changes.
- 3 (Minor mismatch): Most key elements are present, but minor details are missing,
incorrect, or slightly inaccurate.
- 2 (Some mismatch): Some key elements are missing, altered, or interpreted
incorrectly.
- 1 (Major deviations): Key elements are completely missing, altered, or
contradicted. Instruction is ignored.

2. Image Faithfulness (How well are the non-edited parts and key input elements
preserved?)
- 4 (Uses input fully): All relevant elements from the input (background, style,
lighting, identity) are accurately preserved or transformed as instructed.
- 3 (Minor mismatch): Most relevant elements are preserved, but a few aspects (e.g.,
background details, lighting consistency) are missing or incorrectly handled.
- 2 (Partial mismatch): Some elements are carried over, but key aspects of the
original image are lost or distorted.
- 1 (Fails to use input): Key elements of the input image are ignored, misinterpreted,
or destroyed.

3. Physical and Visual Quality (Technical errors, composition, realism, and
physics)
- 4 (No noticeable flaws): The image is physically plausible (correct lighting,
shadows, geometry, anatomy). No visible artifacts (seams, blurring, noise).
- 3 (Minor flaws): Small inaccuracies that are noticeable but not strongly
disruptive (e.g., slight lighting mismatch, minor texture issues).
- 2 (Some flaws): Clear physical or visual errors that disrupt the image (e.g.,
incorrect perspective, ‘‘floating’’ objects, wrong shadow direction, obvious
seams).
```

```
- 1 (Severe flaws): Major physical/visual errors (e.g., impossible geometry,
distorted anatomy, garbled objects, severe artifacts).

4. Text Rendering (Only if the instruction involves generating text)
- 4 (Full match): Text is correct, legible, and integrated well.
- 3 (Mostly match): Minor misspellings or inconsistent capitalization.
- 2 (Partial match): Major misspellings or distorted text.
- 1 (Major deviations): Text is unreadable, severely distorted, or missing. (Use
N/A if no text generation is required).
```

GCR Critique and Refinement Prompt (Part 2)

```
Output your evaluation in the following format:
[ understanding the user request, and what needs to be considered during image
editing ]
# Detailed Judgement
1. Text Faithfulness:
## Score: [ float score ]
## Justification: [Detailed explanation of the score]
2. Image Faithfulness:
## Score: [ float score ]
## Justification: [Detailed explanation of the score]
3. Physical and Visual Quality:
## Score: [ float score ]
## Justification: [Detailed explanation of the score]
4. Text Rendering:
## Score: [ float score or N/A ]
## Justification: [Detailed explanation of the score]
# Summary: [Summary of the evaluation]
# User Request Refinement:
## Refinement Comments: [Explanation of why the original instruction needs
refinement and what constraints should be added]
## Refined Request: [Improved, more specific instruction that addresses identified
deficiencies]
```

GCR Loop Logic. At test time, RationalRewards generates the full critique and refinement output in a single forward pass. If any dimension score falls below the threshold of 3.0, the refined request is extracted and fed back to the generator for re-generation. If all scores are ≥ 3.0, the original generation is accepted. In our experiments, we use a single-iteration loop (i.e., at most one refinement per image).

D Scoring Rubrics

This appendix provides the detailed scoring rubrics for the four assessment dimensions used in pointwise evaluation, referenced in Section 2.1 of the main text. Scores are on a 1–4 integer scale with float-valued interpolation (e.g., 2.5) permitted for fine-grained assessment.

D.1 Text Faithfulness

Evaluates how accurately the generated or edited image follows the text instruction.

Note: Float-valued scores (e.g., 2.5) interpolate between adjacent anchor descriptions to reflect fine-grained quality distinctions.

D.2 Image Faithfulness (Editing Only)

Evaluates how well the edited image preserves elements of the source image that should remain unchanged.

Score	Description
4 (Full match)	All key elements (objects, colors, actions) are represented exactly as described. No hallucinations or unrequested changes.
3 (Minor mismatch)	Most key elements are present, but minor details are missing, incorrect, or slightly inaccurate.
2 (Some mismatch)	Some key elements are missing, altered, or interpreted incorrectly.
1 (Major deviations)	Key elements are completely missing, altered, or contradicted. Instruction is ignored.

Table 7: Scoring rubric for Text Faithfulness.

Score	Description
4 (Uses input fully)	All relevant elements from the input (background, style, lighting, identity) are accurately preserved or transformed as instructed.
3 (Minor mismatch)	Most relevant elements are preserved, but a few aspects (e.g., background details, lighting consistency) are missing or incorrectly handled.
2 (Partial mismatch)	Some elements are carried over, but key aspects of the original image are lost or distorted.
1 (Fails to use input)	Key elements of the input image are ignored, misinterpreted, or destroyed.

Table 8: Scoring rubric for Image Faithfulness.

Scored as N/A for text-to-image generation tasks where no source image is provided.

D.3 Physical and Visual Quality

Evaluates the physical plausibility and overall visual quality of the generated image.

Score	Description
4 (No noticeable flaws)	The image is physically plausible (correct lighting, shadows, geometry, anatomy). No visible artifacts (seams, blurring, noise).
3 (Minor flaws)	Small inaccuracies that are noticeable but not strongly disruptive (e.g., slight lighting mismatch, minor texture issues).
2 (Some flaws)	Clear physical or visual errors that disrupt the image (e.g., incorrect perspective, "floating" objects, wrong shadow direction, obvious seams).
1 (Severe flaws)	Major physical/visual errors (e.g., impossible geometry, distorted anatomy, garbled objects, severe artifacts).

Table 9: Scoring rubric for Physical and Visual Quality.

D.4 Text Rendering

Evaluates the quality and accuracy of any text rendered within the generated image.

Scored as N/A when the instruction does not require text rendering.

E Implementation Details

This appendix provides the training hyperparameters, hardware configuration, and RL algorithm details referenced in Section 3 of the main text.

E.1 RL Fine-Tuning Setup

We employ DiffusionNFT (Zheng et al., 2025) for RL-based parameter-space optimization. DiffusionNFT is an online RL framework that operates on the forward diffusion process via

Score	Description
4 (Full match)	Text is correct, legible, and integrated well into the image.
3 (Mostly match)	Minor misspellings or inconsistent capitalization.
2 (Partial match)	Major misspellings or distorted text.
1 (Major deviations)	Text is unreadable, severely distorted, or missing.

Table 10: Scoring rubric for Text Rendering.

flow matching, avoiding the need for likelihood estimation, solver restrictions, or classifier-free guidance (CFG) required by reverse-process approaches such as FlowGRPO (Xu et al., 2025; Jin et al., 2025; Wu et al., 2025a; Lan et al., 2025; Esser et al., 2024).

Algorithm Overview. DiffusionNFT (Zheng et al., 2025) frames RL for diffusion models as a supervised contrastive learning problem (Wang et al., 2025e;a). At each iteration, the algorithm: (1) samples K images from the current policy for a given prompt; (2) evaluates each image with a reward function; (3) splits images into implicit positive (high-reward) and negative (low-reward) subsets; and (4) updates the model via a contrastive flow-matching loss that pushes the policy toward positive generations and away from negative ones. The key theoretical insight is that the velocity-field difference between positive and negative policies defines a *reinforcement guidance direction* Δ that guarantees policy improvement.

Integration with RationalRewards. RationalRewards produces per-dimension scores (Text Faithfulness, Image Faithfulness, Physical Quality, Text Rendering) for each generated image. We aggregate these into a scalar reward for the DiffusionNFT loss via equal-weight averaging of applicable dimensions (excluding N/A dimensions). Specifically, for a generated image x_0 given prompt c:

$$r(x_0, c) = \frac{1}{|\mathcal{D}_{\text{active}}|} \sum_{d \in \mathcal{D}_{\text{active}}} s_d(x_0, c), \tag{13}$$

where s_d is the score for dimension d and $\mathcal{D}_{\text{active}}$ is the set of applicable dimensions (e.g., excluding Image Faithfulness for T2I tasks and Text Rendering when no text generation is required).

Algorithm 1 provides pseudocode for the RL fine-tuning procedure.

RL Hyperparameters. We employ Low-Rank Adaptation (LoRA) (Hu et al., 2022) for parameter-efficient fine-tuning. Experiments are conducted on a distributed system comprising 16 NVIDIA A100-80GB GPUs, with 8 GPUs dedicated to model training and 8 GPUs serving the reward model for online evaluation. Table 11 summarizes the key hyperparameters.

RL Training Data. We source the RL training prompts from the EditReward Dataset and HPDv3 dataset by selecting prompts whose initial generations receive below-average rewards (mean score < 3.0 from RationalRewards), focusing training on cases where the generator has the most room for improvement (Wang et al., 2025c;f).

E.2 GCR Loop Configuration

At inference time, RationalRewards is served via vLLM with prefix caching and paged attention enabled, achieving a per-image overhead of approximately 0.4 seconds for the full critique-and-refinement pass. The refinement threshold is set to 3.0: if any dimension score falls below this value, the refined prompt is used for re-generation. This threshold was selected as the midpoint of the 1–4 scoring scale, corresponding to the boundary between "minor issues" (score 3) and "notable deficiencies" (score 2) in our rubrics (Appendix D).

Require: Flow model v_θ, reference policy $v_{\text{old}} \leftarrow v_\theta$, RationalRewards model $\mathcal{R}$, prompt dataset $\mathcal{C}$, group size K, guidance strength β, EMA schedule $\{\eta_i\}$, number of iterations N

for iteration $i = 1, \ldots, N$ **do**
// Phase 1: Online Data Collection
Sample batch of prompts $\{c_j\}_{j=1}^{B}$ from $\mathcal{C}$
for each prompt c_j **do**
Generate K images $\{x_0^{(k)}\}_{k=1}^{K}$ using current sampling policy v_{old}
Compute raw rewards: $r_{\text{raw}}^{(k)} \leftarrow \mathcal{R}(x_0^{(k)}, c_j)$ *// Multi-dim scores aggregated via Eq. (D.1)*
Normalize rewards within group: $r^{(k)} \leftarrow 0.5 + 0.5 \cdot \text{clip}\left(\frac{r^{(k)} - \bar{r}}{Z_c}, -1, 1\right)$
Store $\{c_j, x_0^{(1:K)}, r^{(1:K)}\}$ in buffer $\mathcal{D}$
end for
// Phase 2: Policy Optimization (Forward Process)
for each $(c, x_0, r) \in \mathcal{D}$ **do**
Sample timestep $t \sim \mathcal{U}(0, 1)$ and noise $\epsilon \sim \mathcal{N}(0, I)$
Compute noisy image: $x_t \leftarrow \alpha_t x_0 + \sigma_t \epsilon$
Compute flow-matching target: $v \leftarrow \alpha_t' x_0 + \sigma_t' \epsilon$
Compute implicit positive velocity: $v_\theta^+ \leftarrow (1 - \beta) v_{\text{old}}(x_t, c, t) + \beta \cdot v_\theta(x_t, c, t)$
Compute implicit negative velocity: $v_\theta^- \leftarrow (1 + \beta) v_{\text{old}}(x_t, c, t) - \beta \cdot v_\theta(x_t, c, t)$
Compute loss: $\mathcal{L} \leftarrow r \cdot \|v_\theta^+ - v\|^2 + (1 - r) \cdot \|v_\theta^- - v\|^2$
end for
Update θ via gradient descent on $\mathcal{L}$
// Phase 3: Soft EMA Update of Sampling Policy
$\theta_{\text{old}} \leftarrow \eta_i \theta_{\text{old}} + (1 - \eta_i)\theta$
end for
returnFine-tuned model v_θ

Algorithm 1: RL Fine-Tuning with RationalRewards via DiffusionNFT

Hyperparameter	**Value**
Resolution	512×512
Guidance Scale (Flux.1 Kontext Dev)	2.5
Sampling Steps (Training)	15 (DPM solver)
Sampling Steps (Evaluation)	20
Noise Level	0.7
Learning Rate	2e-4
β (guidance strength)	0.0001
Batch Size (per GPU)	8
Group Size K	16 (across 16 process groups)
Quality Filtering (mean threshold)	0.9
Quality Filtering (std threshold)	0.05
LoRA Rank	64
LoRA Alpha	128
GPU Configuration	$8 \times$ A100-80GB (training) + $8 \times$ A100-80GB (reward)
Training Wall-Clock Time	$\sim$16 GPU-hours per generator

Table 11: Hyperparameters for RL fine-tuning via DiffusionNFT.

F Dataset and Benchmark Details

F.1 Training Data Statistics

Table 12 provides detailed statistics for the training data used in the PARROT pipeline (Section 3, main text).

Source Dataset	Task	Raw Pairs	Post-Filtering Pairs	Final Pointwise Samples
EditReward	Image Editing	30K	∼21.6K	∼43.2K
HPDv3	Text-to-Image	50K	∼36K	∼55K
RapidData	Text-to-Image			

Table 12: Training data composition before and after consistency filtering. Each pairwise sample yields two pointwise projection samples (one per image).

We note that our total training scale (∼80K raw pairs, ∼57.6K after filtering) is substantially smaller than comparable baselines: EditReward uses 200K pairs and UnifiedReward uses over 1M pairs. Part of this data efficiency stems from the teacher model's pre-trained knowledge, which PARROT distills through structured rationales rather than raw labels.

F.2 Consistency Filtering Analysis

The consistency filtering step (Phase 2, Section 2.1) retains approximately 72% of generated rationales overall. We observe the following common failure modes in rejected rationales:

1. **Visual hallucination**: The teacher generates a rationale describing visual content not present in the images (e.g., "Image A contains a clear sunset in the background" when no sunset is visible), leading to an incorrect preference prediction when the label hint is removed.
2. **Label-ignoring rationales**: Despite the preference anchor, the teacher occasionally generates a rationale that favors the non-preferred image, particularly when the quality difference between images is subtle.
3. **Vague, non-predictive reasoning**: The rationale provides generic praise or criticism (e.g., "Both images are of reasonable quality") without sufficient discriminative detail to distinguish between the two options.

F.3 Evaluation Benchmark Summary

Table 13 summarizes all evaluation benchmarks used in this work.

Benchmark	Task	# Samples	Evaluation Protocol	Metrics	Reference
MMRB2 (T2I)	Preference Prediction	1000	Pairwise comparison	Accuracy	Hu et al. (2025)
MMRB2 (Edit)	Preference Prediction	1000	Pairwise comparison	Accuracy	Hu et al. (2025)
EditReward Bench	Preference Prediction	133	Pairwise comparison	Accuracy	Wu et al. (2025e)
GenAI-Bench (T2I)	Preference Prediction	1700	Pairwise comparison	Accuracy	Jiang et al. (2024)
GenAI-Bench (Edit)	Preference Prediction	900	Pairwise comparison	Accuracy	Jiang et al. (2024)

Table 13: Summary of evaluation benchmarks.

G Limitations and Broader Impact

G.1 Limitations

We acknowledge the following limitations of this work:

1. **Teacher Model Dependence.** The quality of RationalRewards is upper-bounded by the teacher model (Qwen3-VL-32B-Instruct) used to generate training rationales. In

domains where the teacher exhibits systematic blind spots—such as fine-grained physics simulation, culturally specific aesthetics, or specialized technical content—the student model inherits these limitations. Future work could explore ensembling multiple teacher models or incorporating human-in-the-loop corrections for high-stakes domains.

2. **Bias Inheritance.** Preference datasets (EditReward, HPDv3, RapidData) encode the aesthetic preferences and cultural assumptions of their annotators. The teacher VLM introduces additional biases from its own pretraining data. RationalRewards may therefore systematically favor certain visual styles, demographics, or content types. We have not conducted a comprehensive bias audit, and we encourage users to evaluate the model's behavior on diverse and potentially underrepresented content before deployment.
3. **Latent Capability Hypothesis.** Our finding that test-time prompt tuning matches or exceeds RL-based fine-tuning (Section 3.2) supports the hypothesis that generators harbor latent capabilities under-elicited by suboptimal prompts. However, this remains a working hypothesis: we have not validated it at the representation level (e.g., by probing internal activations), and alternative explanations—such as the prompt refinement simply providing additional context that any model would benefit from—cannot be ruled out.
4. **Threshold Sensitivity.** The GCR loop uses a fixed threshold of 3.0 to trigger refinement. While this corresponds to a natural boundary in our scoring rubric (Appendix D), we have not conducted a comprehensive sensitivity analysis across all benchmarks and generators. The optimal threshold may vary by generator capability and task difficulty.
5. **Language and Domain Scope.** All evaluation in this work is conducted on English-language benchmarks. The transferability of RationalRewards' structured critiques to other languages, as well as to non-photorealistic domains (e.g., 3D rendering, video generation, scientific visualization), remains untested.

G.2 Broader Impact

RationalRewards and the PARROT framework contribute to the growing ecosystem of tools for evaluating and improving visual generation. We anticipate both positive and negative societal implications:

Positive impacts.

- **Democratized evaluation:** By providing an open-source, reasoning-based reward model competitive with commercial alternatives, we lower the barrier for researchers and practitioners to evaluate visual generation quality without relying on costly proprietary APIs.
- **Interpretability:** Structured, multi-dimensional critiques provide transparent explanations for quality assessments, enabling users and developers to understand and address specific failure modes rather than optimizing against opaque scalar scores.
- **Accessibility:** The GCR loop can help users with limited prompt engineering experience achieve higher-quality generations by automatically identifying and addressing deficiencies in their instructions.

Negative impacts and mitigations.

- **Misuse potential:** Improved image generation quality could be leveraged for creating misleading visual content, deepfakes, or other harmful media. We note that RationalRewards itself does not generate images but evaluates and critiques them; however, its use as an RL reward or prompt optimizer could amplify generator capabilities.
- **Bias amplification:** As discussed in the limitations, reward models trained on biased preference data may systematically favor certain content types, potentially amplifying existing disparities in visual representation.
- We encourage responsible use and recommend that practitioners conduct domain-specific evaluations before deploying RationalRewards in production systems, particularly in sensitive applications.

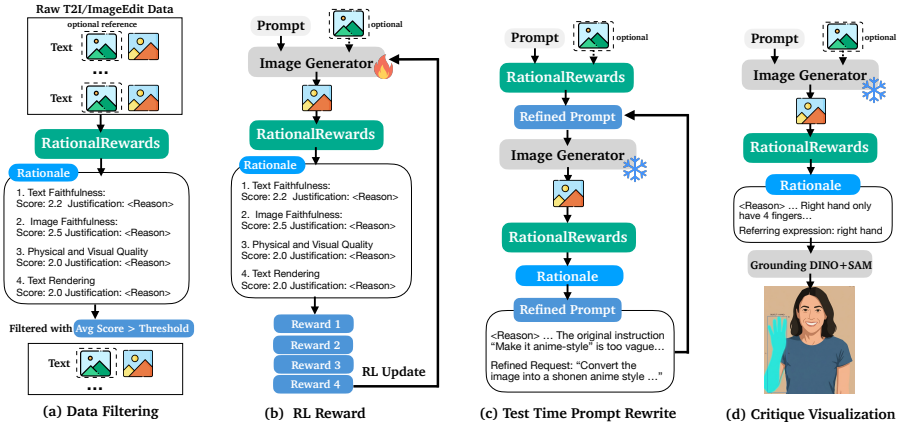


Figure 13: `RationalRewards` (a) enables explainable quality control for data curation; (b) serves as a multi-dimensional reward model driven by transparent rationales; (c) serves as a preference-calibrated test-time prompt tuner that trades compute for better generation quality; (d) fuels regional flaw grounding and dense visual rewards.

H Additional Applications and Diagnostic Uses

Beyond the optimization results in the main paper, `RationalRewards` supports rationale-based data filtering, test-time prompt rewriting, and regional critique grounding.