

SourceReward: Source-Preserving Reward Modeling for High-Precision Image Editing

Rishav Sahay*
rissahay@amazon.com

Karan Malhotra*
kamlhtra@amazon.com

Lavanya Tekumalla
lavanya@amazon.com

Akanksha Roy†
akankshar610@gmail.com

Purav Aggarwal
aggap@amazon.com

Amazon

* Equal contribution (first authors).

† Work done during an internship at Amazon.

Abstract

Instruction-guided image editing models excel at following instructions but often introduce unintended changes to source regions that should remain unchanged, a critical failure in high-precision applications like product troubleshooting. Existing reward models primarily evaluate instruction adherence and perceptual quality, but do not explicitly assess whether edits preserve the underlying source content outside the intended modification. We introduce **SourceReward**, a reward model that incorporates source adherence as a first-class evaluation dimension. Our approach jointly reasons over the source image, edit instruction, and edited output, using edit-centric cross-attention to capture both source consistency and instruction alignment, enabling the model to distinguish intended modifications from unintended deviations. To address the lack of human preference data in domain-specific settings, we propose a scalable cold-start weak-supervision pipeline that generates diverse edit candidates, aggregates multi-judge evaluations via Reciprocal Rank Fusion, and leverages inter-judge agreement for uncertainty-aware training. Experiments across three editing datasets show that SourceReward improves best-of-N selection and accept/reject filtering, complementing standard pairwise preference evaluation with deployment-oriented metrics. Cross-domain evaluation across product troubleshooting, EditReward, and MagicBrush datasets shows that SourceReward improves ROC-AUC over the EditReward baseline by 35.7%, 8.7%, and 29.9% respectively, while improving SuccessRate@1 by 26.5% on product troubleshooting. SourceReward also reduces per-sample scoring latency by $15.6\times$ relative to VLM-based judging.

1 Introduction

With the rapid advancement of image generation models, reward models for image generation have become indispensable, both as quality filters for selecting among generated

candidates and as training signals for RLHF-based fine-tuning [10] and synthetic data curation. Models such as ImageReward [21] and HPS [19] have demonstrated substantial gains in text-to-image synthesis through best-of- N re-ranking and preference learning. A natural frontier is instruction-guided image editing, where models such as QwenImageEdit [14], InstructPix2Pix [9], MagicBrush [27], and SeedEdit 3.0 [15] enable rich modifications from natural language.

Reward models tailored to editing are just emerging: EditReward [20] learns human-aligned preference scores, Multi-Reward as Condition [8] decomposes quality into conditioning signals, and EditScore [6] scales reward-guided online RL to 72B-parameter models. Yet these efforts share a critical blind spot. They primarily evaluate whether the edit *follows the instruction* and whether the output *looks realistic*, while largely ignoring whether the edited image *faithfully preserves the source content* in regions that should remain unchanged. We term this overlooked dimension *source adherence*: the degree to which non-edited regions maintain structural and visual fidelity to the original image. Such failures are pervasive—resizing an object can destroy surrounding content, and adding overlays can corrupt the underlying structure (Figure 1). While tolerable in casual creative editing, these artifacts become consequential in precision-critical applications such as multimodal product troubleshooting [18, 22, 26], where edits must preserve device structure exactly.

The most prominent standalone reward model for scoring image edits is EditReward [20], which conditions on the source image, instruction, and edited output through joint encoding. However, preservation of non-edited regions is implicitly bundled within instruction-following signals rather than modeled as a distinct objective, making it difficult to distinguish edits that satisfy the instruction but introduce unintended changes elsewhere. Moreover, joint encoding does not enforce explicit correspondence between edited regions and their source or instruction context. We address this by introducing an edit-centric formulation with cross-attention that explicitly compares edited regions against both the source and instruction.

This gap is not only architectural but also evaluative. Existing editing reward models are commonly assessed through pairwise preference alignment, which is useful for reward-guided training, RLHF-style optimization, and synthetic data curation. However, deployment pipelines for image editing require additional decision capabilities: a generated edit may need to be accepted or rejected in isolation, ranked among multiple candidates in a best-of- N setting, or rejected despite high visual plausibility because it introduces subtle source-inconsistent changes. These use cases require calibrated, reference-grounded signals rather than only relative pairwise preferences. We therefore evaluate our reward model not only with pairwise accuracy, but also with deployment-oriented ranking and classification metrics that measure whether source-preserving edits can be selected in best-of- N settings and accepted or rejected in isolation.

Finally, addressing this gap requires not only a new modeling formulation but also a practical solution to the *cold-start problem* in high-precision editing domains. In product troubleshooting, for example, source adherence failures can be operationally consequential: a misplaced port, missing button, or altered device layout may directly mislead users. Yet such specialized domains rarely have labeled edited images, human rankings, or curated preference benchmarks at inception. Building a reward model in this regime therefore requires jointly constructing edit candidates and preference signals. We instantiate this setting in product troubleshooting and evaluate SourceReward across three editing datasets [21], including human-labeled EditReward data, to assess both domain-specific reliability and broader generalization.

A summary of our contributions is as follows.

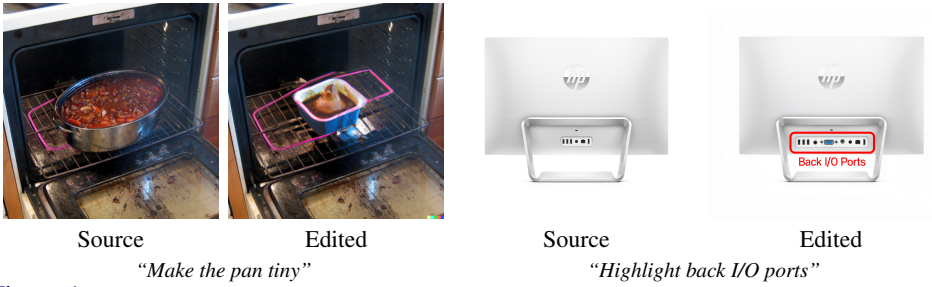


Figure 1: Sample edits violating source adherence. Left pair: resizing the pan distorts surrounding objects and destroys the content inside the pan. Right pair: the edit correctly labels the I/O ports but distorts the set of available ports.

- We introduce a reward formulation that elevates source adherence to a first-class objective alongside instruction relevance and visual integrity, addressing a fundamental gap in existing evaluation approaches that conflate intended edits with unintended source deviations.
- We propose an edit-centric reward model that jointly conditions on the source image, edit instruction, and edited output using cross-attention to capture edited-to-source consistency and edited-to-instruction alignment, and train it with a hybrid pointwise–pairwise objective with uncertainty-aware weighting.
- We develop a scalable cold-start weak-supervision pipeline for annotation-scarce editing domains, generating diverse edit candidates and aggregating ensemble VLM judgments via Reciprocal Rank Fusion, with inter-judge agreement used as a confidence signal for training.
- We evaluate SourceReward across product troubleshooting, EditReward, and MagicBrush datasets using deployment-oriented metrics. On product troubleshooting, SourceReward improves over EditReward baseline by +26.5% SuccessRate@1, +35.7% ROC-AUC, and +23.9% F1, while reducing scoring latency by $15.6\times$ relative to VLM judging and $5.0\times$ relative to EditReward baseline.

2 Related Work

Recent advances in instruction-guided image editing build on models such as QwenImageEdit [10], InstructPix2Pix [11], MagicBrush [12], and SeedEdit 3.0 [13], and use test-time scaling to generate multiple candidates and select outputs using reward models or verifiers [14, 15, 16, 17, 18, 19]. This makes reward design central to reliable editing.

2.1 Reward Models for Image Generation and Editing

Reward modeling for visual generation originates in text-to-image synthesis, where models such as ImageReward [20] and HPS [21] learn human preference scores from large-scale annotations for prompt–image alignment, aesthetic quality, RLHF-based fine-tuning [22], and best-of- N reranking. Classical metrics such as FID [23], CLIPScore [24], and LPIPS [25]

capture distributional or perceptual similarity, but fail to provide fine-grained, human-aligned assessment for editing tasks.

Reward modeling for instruction-guided image editing is more recent. EditReward [17] learns human-aligned scores conditioned on the source image, instruction, and edited output, but relies on joint encoding without isolating preservation of non-edited regions. Multi-Reward as Condition [8] decomposes quality into multiple signals but uses coarse preservation measures for conditioning. EditScore [8] scales reward models via online reinforcement learning to improve instruction adherence, while SeedEdit 3.0 [18] focuses on editing architectures rather than reward design.

Recent evaluation frameworks highlight limitations of existing methods. ComplexBench-Edit [14] introduces a vision-consistency metric for non-modified regions but remains a fixed metric rather than a learned reward. More broadly, prior work emphasizes improved evaluation protocols [6], statistical assessment of synthetic images [25], and robustness to annotation noise [24]. Building on these insights, we learn a reference-grounded reward model that explicitly models source adherence alongside instruction relevance and visual integrity.

2.2 Cold-Start Preference Learning for Image Editing

A key challenge in domain-specific image editing is the absence of human preference data. Unlike general-purpose text-to-image generation [21], specialized settings such as product troubleshooting lack curated datasets of edited images and annotations. Prior work highlights challenges of noisy supervision in visual evaluation [24], motivating approaches based on synthetic data generation and automated scoring.

Recent methods construct preference signals by generating multiple candidates and filtering them using learned reward models [7, 9, 10, 11, 13, 16]. However, these approaches assume reliable reward signals. Our work addresses this gap by jointly designing a reward model and a cold-start data construction pipeline that leverages ensemble evaluation and uncertainty-aware aggregation.

2.3 Multimodal RAG and Visual Content Generation for Product Troubleshooting

Recent multimodal retrieval-augmented generation (RAG) systems combine retrieved context with generative multimodal outputs [18, 22, 23, 26]. In practical product troubleshooting assistants, responses are constructed from structured knowledge bases (KBs), where retrieved product images are edited to illustrate diagnostic steps such as highlighting ports or buttons. Prior work such as MIRAGE [12] focuses on metadata-guided retrieval of existing images but does not address generating images through editing.

Such editing systems require selecting a single output from multiple generated candidates, making reward-based filtering critical. While pairwise judging approaches [7] are effective for relative ranking, they lack calibrated, reference-grounded signals needed for reliable top-1 selection, especially when errors arise from subtle structural deviations. Our work addresses this gap by enabling efficient and accurate filtering for high-fidelity multimodal RAG deployment.

3 SourceReward for High-Precision Image Editing

We consider instruction-guided image editing, where a source image I_s and instruction t produce an edited image I_e . Our goal is to learn a reward model $R(I_s, t, I_e)$ that evaluates edits in a reference-grounded setting, explicitly distinguishing intended modifications from unintended deviations. An overview of our framework is shown in Figure 2.

Existing reward models primarily assess instruction adherence and perceptual quality, but do not explicitly capture whether non-edited regions remain faithful to the source image. We address this by introducing a source-preserving reward formulation and an edit-centric architecture that jointly reasons over source image, instruction, and edited output.

3.1 Source-Preserving Reward Formulation

Unlike prior approaches that rely on joint encoding, we explicitly model correspondence between the edited image, source image, and instruction using cross-attention. We decompose edit quality into three complementary components: instruction relevance R_{instr} , source adherence R_{src} , and visual integrity R_{vis} . R_{instr} measures whether the edit follows the instruction, R_{src} evaluates whether non-edited regions remain consistent with the source image, and R_{vis} captures perceptual quality.

$$R(I_s, t, I_e) = w_1 R_{\text{instr}} + w_2 R_{\text{src}} + w_3 R_{\text{vis}} \quad (1)$$

This decomposition is critical: it separates *intended changes* (captured by R_{instr}) from *unintended deviations* (captured by R_{src}), which are typically conflated in prior reward formulations.

3.2 Edit-Centric Cross-Attention Modeling

To evaluate edits relative to both the source and instruction, we model the edited image as the primary object of evaluation and condition it on both signals (Figure 2, top). We encode the source image, edited image, and instruction into token representations S , E , and T using a shared vision-language encoder.

We construct two complementary representations using cross-attention:

$$H_{\text{src}} = \text{CrossAttn}(Q = E, K = S, V = S), \quad (2)$$

$$H_{\text{txt}} = \text{CrossAttn}(Q = E, K = T, V = T). \quad (3)$$

Here, the edited image acts as the query in both cases. H_{src} captures edited-to-source consistency, while H_{txt} captures alignment with the instruction. This asymmetric design enforces an *edit-centric* perspective: each region of the edited image is evaluated based on (i) what it was in the source and (ii) whether it should have changed. We fuse these representations as $H = \text{Proj}([E; H_{\text{src}}; H_{\text{txt}}])$, and pass pooled features to three task-specific heads to predict R_{instr} , R_{src} , R_{vis} . This enables the model to distinguish between correct localized edits, unchanged regions, and unintended deviations.

Multi-head reward prediction and aggregation. From the fused representation H , we predict three reward components R_{instr} , R_{src} , R_{vis} using task-specific heads. The final reward is computed as a weighted combination of these components (Eq. 1). We learn aggregation

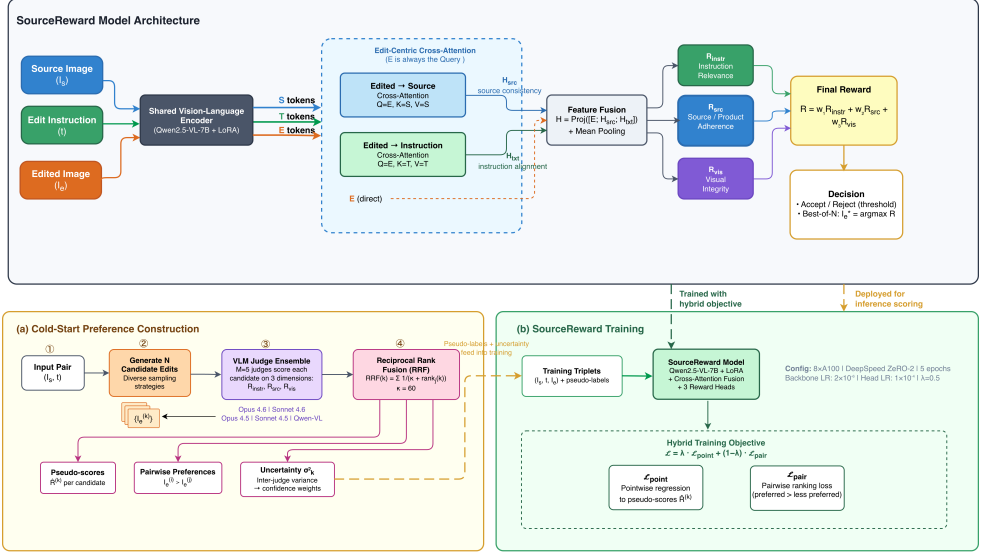


Figure 2: Overview of the SourceReward framework. **Top:** Model architecture (shared across training and inference) showing the shared vision-language encoder, edit-centric cross-attention (with edited-image tokens E as query), feature fusion, and three reward heads (R_{instr} , R_{src} , R_{vis}). **Bottom-left:** Cold-start preference construction pipeline generating pseudo-labels via a VLM judge ensemble and Reciprocal Rank Fusion. **Bottom-right:** Hybrid training objective combining pointwise regression and pairwise ranking with uncertainty-aware weighting derived from inter-judge variance σ_k^2 .

weights dynamically using a lightweight MLP that takes the three per-head scalar scores as input and predicts softmax-normalized weights: $\mathbf{w} = \text{softmax}(\text{MLP}([R_{instr}, R_{src}, R_{vis}]))$, where the MLP is a two-layer network ($3 \rightarrow 128 \rightarrow 3$) with ReLU activation.

3.3 Training and Inference

We train the reward model using a hybrid objective that combines pointwise scoring with robust ranking under noisy supervision. This design intentionally supports both RLHF-style preference learning and deployment-time accept/reject or best-of-N selection.

Pointwise supervision: We regress the model’s predicted reward to pseudo-scores derived from aggregated judge evaluations, encouraging the model to produce pointwise absolute quality estimates:

$$\mathcal{L}_{\text{point}} = \sum_k \alpha_k \left(R(I_s, t, I_e^{(k)}) - \hat{R}^{(k)} \right)^2, \quad (4)$$

where $\alpha_k = 1 / (1 + \sigma_k^2)$ and σ_k^2 is the inter-judge score variance for sample k .

Pairwise supervision: We additionally enforce that preferred edits receive higher scores than less preferred ones, improving ranking consistency across candidates:

$$\mathcal{L}_{\text{pair}} = \sum_{(i,j)} \beta_{ij} \log(1 + \exp(-(R_i - R_j))). \quad (5)$$

where $\beta_{ij} = \min(\alpha_i, \alpha_j)$ reflects the confidence in the preference pair. Final objective is obtained as $\mathcal{L} = \lambda \mathcal{L}_{\text{point}} + (1 - \lambda) \mathcal{L}_{\text{pair}}$. The two objectives are complementary: pointwise learning provides scores for filtering decisions, while pairwise learning improves relative ranking for candidate selection. Both are weighted by inter-judge agreement to account for label uncertainty.

Inference: At inference time, SourceReward supports both single-candidate accept/reject decisions with a threshold and multi-candidate ranking via $I_e^* = \arg \max_{I_e} R(I_s, t, I_e)$, motivating our evaluation using both top-1 and ranking-based metrics (details in Sec. B of the supplementary material).

3.4 Cold-Start Preference Construction

We operate in a setting without human preference annotations. For each (I_s, t) pair, we generate N candidate edits $\{I_e^{(k)}\}$ using diverse sampling strategies. Specifically, we employ two complementary generation backends: Qwen-Edit (an instruction-following image editor) and FLUX (a diffusion-based generator), each invoked with multiple random seeds (v0, v1, v2), yielding $N=6$ candidates per sample that vary in both model inductive bias and stochastic realization. An ensemble of $M=5$ VLMs jointly evaluates all candidates, producing scalar scores and rankings across the three reward dimensions (Figure 2, bottom-left). Each judge independently scores every candidate on prompt relevance, source adherence, and visual integrity on a 1-10 comparative scale, per-metric scores are averaged into an overall quality score from which each judge derives its own ranking. We aggregate rankings using Reciprocal Rank Fusion to obtain pseudo-scores $\hat{R}^{(k)}$ and pairwise preferences $(I_e^{(i)} \succ I_e^{(j)})$. Inter-judge agreement (measured as rank variance across models) serves as a confidence signal, reflecting ambiguity in evaluation.

$$\text{RRF}(k) = \sum_{j=1}^M \frac{1}{\kappa + \text{rank}_j(k)}, \quad (6)$$

4 Experiments

We evaluate SourceReward along four questions: (i) does explicitly modeling source adherence improve high-precision edit selection, (ii) are the gains consistent across multiple editing datasets, (iii) can SourceReward support inference-time decisions such as best-of- N selection and accept/reject filtering, and (iv) can cold-start judge supervision align with human-labeled editing preferences. To answer these questions, we evaluate using ranking, classification, and pairwise preference metrics across three editing datasets.

4.1 Experimental Setup

Datasets. We evaluate on one proprietary and two public editing datasets, summarized in Table 1. Candidate edits and pseudo-labels for Product Troubleshooting and MagicBrush are

Dataset	Type	Images	Train	Val	Test
Product Troubleshooting	Proprietary	1,578	32,693	1,912	2,006
EditReward [20]	External	8,132	50,854	2,617	1,851
MagicBrush [24]	External	1,053	8,807	1,053	528

Table 1: Datasets used for SourceReward training and evaluation. Counts are pairwise preferences for Product Troubleshooting and EditReward, and edit triples for MagicBrush. Best-of- N ranking is evaluated on test groups of 6 candidates per source–instruction pair (366 groups for Product Troubleshooting, 254 for EditReward).

generated via the cold-start pipeline in Sec. 3.4. For EditReward, we use its existing pool of 6 pre-generated candidate edits per query directly, applying only the pseudo-labeling stage. Across all three datasets, the pipeline successfully labels 54,111 triplets in total.

Product Troubleshooting (proprietary) is our primary high-precision setting, covering device-specific edits such as highlighting ports, labeling buttons, and annotating device layouts, where source-adherence failures are operationally consequential. The dataset comprises 1,578 source images and 32,693 training preference pairs, with 2,006 test pairs organized into 366 best-of- N groups of 6 candidates per source–instruction query. Pseudo-label quality is validated against 500 held-out expert human judgments in Sec. 4.2.

EditReward [20] provides human-labeled instruction-guided editing examples with per-image annotator scores covering instruction-following and visual quality. The dataset contains 8,132 source images and 50,854 training preference pairs, with 1,851 test pairs in 254 best-of- N groups. Since candidate edits are already provided, we skip image generation and apply only pseudo-labeling. Each example carries an aggregated human overall quality score, serving as a ground-truth probe of model quality against 1,577 non-tied held-out human judgments in Sec. 4.3 (Table 7).

MagicBrush [24] is a general instruction-guided editing benchmark covering object addition, removal, and attribute modification, consisting of 1,053 source images, 8,807 training triples, and 528 test triples. We include it to test generalization beyond product-centric imagery under the same cold-start protocol.

Judge ensemble: The pseudo-labels are produced using five open-source vision-language judges drawn from the Qwen and Llama model families, spanning 11B to 235B parameters. Each judge scores candidate edits along three dimensions: instruction relevance, source/product adherence, and visual integrity. Rankings are aggregated using Reciprocal Rank Fusion (RRF) with $\kappa = 60$.

Evaluation metrics: We evaluate reward models under three usage modes that correspond to common inference-time decisions in image editing. For *preference-learning* quality, we report *pairwise accuracy* over candidate pairs. For *best-of- N selection*, a reward model ranks multiple candidate edits generated for the same source image and instruction; we report *NDCG@6* and *SuccessRate@1* (SR@1), where SR@1 measures whether the model’s top-ranked selection is an acceptable edit, defined as scoring above a quality threshold on all three dimensions (prompt relevance, source fidelity, and visual integrity). For *single-candidate accept/reject filtering*, the model must decide whether an edit is acceptable in isolation; we report *ROC-AUC* and *macro F1*. These metrics complement standard pairwise reward-model evaluation. Pairwise accuracy is useful for preference learning and reward-guided optimization, but inference-time editing systems must also select a single output from a candidate set and reject low-quality edits. We therefore report both ranking and

Model	Mean	Std	$\hat{R}_{\text{instr}}$	$\hat{R}_{\text{src}}$	$\hat{R}_{\text{vis}}$
Llama-V (11B)	6.52	1.31	6.24	6.81	6.59
Qwen-VL (72B)	6.77	1.46	6.28	7.03	6.96
Llama-V (90B)	6.89	1.10	6.52	7.16	6.96
Llama-4 (109B)	7.53	1.16	6.53	8.21	7.84
Qwen-VL (235B)	8.01	1.45	6.91	8.70	8.25

Table 2: Per-model scoring statistics across 54,111 successfully labeled triplets. $\hat{R}_{\text{instr}}$, $\hat{R}_{\text{src}}$, $\hat{R}_{\text{vis}}$ are mean per-dimension scores per judge. Parenthetical values denote each open-source judge’s public parameter count.

filtering metrics. For all datasets, relative improvements are computed with respect to EditReward ZS within the same dataset; for the proprietary product troubleshooting dataset, absolute scores are omitted due to confidentiality.

Baselines: We compare against two classes of baselines. First, general reward and frozen-encoder baselines: HPSv3, a learned text–image preference model, and **OTS-3D** (Off-The-Shelf 3-Dimensional ensemble), a frozen-encoder baseline we construct by scoring each rubric dimension with a fixed pretrained encoder: prompt relevance via CLIP ViT-L/14 [4] cosine similarity, visual integrity via PickScore-v1, and source adherence via DINOv2-base [28] [CLS] cosine similarity between source and edit. Second, we compare against EditReward, a recent human-aligned reward model for instruction-guided image editing.

SourceReward variants: We evaluate four SourceReward variants as a progressive ablation. **SR-PT+MLP** uses pointwise training with learned aggregation weights. **SR-PT+VarW** adds variance-based uncertainty weighting to the loss term. **SR-PT+PR+VarW** further adds pairwise ranking supervision on top of pointwise training. **SR-CA+PT+PR+VarW** is our full model, which additionally incorporates edit-centric cross-attention for explicit source and instruction grounding.

Implementation details: All SourceReward variants are built on Qwen2.5-VL-7B-Instruct, with the language modeling head replaced by task-specific reward heads. We fine-tune the backbone with LoRA (rank 256, $\alpha=512$) and train cross-attention and reward head parameters in full precision. We train for 5 epochs on 8 GPUs using DeepSpeed ZeRO-2 with effective batch size 64, backbone learning rate 2×10^{-6} (cosine schedule, 5% warmup), and a separate learning rate of 1×10^{-4} for reward heads and cross-attention. Full architecture and training details are provided in the supplementary material (Appendix C).

4.2 Cold-Start Signal Characteristics

Table 2 shows scoring behavior across the five VLM judges. The models differ substantially in score calibration: the largest judge (Qwen-VL 235B) assigns the highest average scores, while the smallest (Llama-V 11B) is more conservative. This variability motivates rank level aggregation rather than relying on raw scalar scores. Inter-judge agreement is moderate rather than high, with exact rank agreement ranging from 27.3% to 50.0%. This level of disagreement motivates the use of ensemble aggregation and uncertainty-aware weighting instead of relying on any single VLM judge.

Across dimensions, instruction relevance consistently receives lower scores than source adherence and visual integrity, indicating that executing the requested edit is often more chal-

Dimension	Accuracy	F1	Precision	Recall
Prompt Relevance	0.688	0.688	0.647	0.733
Product Adherence	0.802	0.871	0.853	0.889
Visual Integrity	0.729	0.787	0.828	0.750
Overall	0.719	0.752	0.804	0.707

Table 3: Cold-start pseudo-labels vs human judgments (binary threshold 7).

Table 4: Pairwise preference accuracy across editing datasets. For product troubleshooting, values are percentage changes over EditReward ZS; the EditReward ZS absolute value is omitted due to confidentiality. For EditReward and MagicBrush, the EditReward ZS absolute score is shown, and all other values are percentage changes over EditReward ZS within the same dataset. Higher is better.

Method	Product Troubleshooting	EditReward	MagicBrush
OTS-3D	−16.0%	−30.4%	−7.3%
HPSv3	−16.3%	−25.4%	+10.2%
EditReward ZS (2 heads)	–	0.889	0.697
EditReward FT (2 heads)	+2.3%	+1.7%	+1.3%
SR-PT+MLP	+3.7%	+1.3%	+24.4%
SR-PT+VarW	+3.3%	+1.9%	+25.5%
SR-PT+PR+VarW	+4.9%	+2.2%	+26.1%
SR-CA+PT+PR+VarW	+3.3%	+5.1%	+27.5%

lenging than preserving the source image or maintaining visual plausibility. This difference in score distributions suggests that the three dimensions capture distinct failure modes and motivates explicitly modeling them rather than collapsing them into a single scalar reward.

A deeper analysis of per-dimension variance (Appendix Table 3 in the supplementary material) reveals that source adherence exhibits the highest inter-judge variance, confirming it is the most ambiguous dimension. Meanwhile, instruction relevance accounts for the majority of low-score samples (17.7% below threshold 7), while source adherence and visual integrity failures are rarer (2.9% and 3.0%). This confirms that different dimensions capture distinct error modes, providing a useful learning signal despite label noise.

Cold-start pipeline vs. in-domain humans: To validate pseudo-label quality, we compare cold-start pipeline scores against expert human judgments on 500 held-out product-troubleshooting examples, binarizing both at threshold 7 (the boundary between ‘partially acceptable’ and ‘acceptable’ for customer-facing deployment). The pipeline achieves 80.4% precision and 70.7% recall overall, with product adherence reaching 88.9% recall (Table 3), showing the cold-start pipeline reliably identifies acceptable edits.

4.3 Main Results: Reward Modeling Performance across Datasets

Tables 4, 5, and 6 summarize performance across three evaluation modes: pairwise preference prediction, best-of- N candidate selection, and single-candidate accept/reject filtering. On pairwise preference accuracy (Table 4), SourceReward improves over EditReward ZS across all datasets, with gains of +3.3% on Product Troubleshooting, +5.1% on EditReward,

Table 5: Best-of- N ranking performance across editing datasets. For product troubleshooting, values are percentage changes over EditReward ZS; the EditReward ZS absolute values are omitted due to confidentiality. For EditReward and MagicBrush, the EditReward ZS absolute scores are shown, and all other values are percentage changes over EditReward ZS within the same dataset. SR@1 = SuccessRate@1. Higher is better.

Method	Product Troubleshooting		EditReward		MagicBrush	
	NDCG@6	SR@1	NDCG@6	SR@1	NDCG@6	SR@1
OTS-3D	-0.2%	-0.5%	-5.5%	-26.0%	-1.5%	-12.5%
HPSv3	-1.2%	-10.5%	-9.5%	-45.3%	+2.5%	+3.5%
EditReward ZS	—	—	0.920	0.799	0.890	0.575
EditReward FT	+4.2%	+25.3%	+0.4%	+1.5%	+3.0%	+5.3%
SR-PT+MLP	+5.1%	+23.1%	-0.3%	+1.0%	+5.2%	+24.0%
SR-PT+VarW	+5.1%	+27.3%	+0.3%	+1.5%	+4.6%	+25.2%
SR-PT+PR+VarW	+5.2%	+24.8%	-0.3%	+1.0%	+4.9%	+27.7%
SR-CA+PT+PR+VarW	+5.4%	+26.5%	+1.4%	+6.8%	+5.1%	+28.0%

Table 6: Accept/reject filtering performance across editing datasets. For product troubleshooting, values are percentage changes over EditReward ZS; the EditReward ZS absolute values are omitted due to confidentiality. For EditReward and MagicBrush, the EditReward ZS absolute scores are shown, and all other values are percentage changes over EditReward ZS within the same dataset. AUC = ROC-AUC; F1 = macro F1. Higher is better.

Method	Product Troubleshooting		EditReward		MagicBrush	
	AUC	F1	AUC	F1	AUC	F1
OTS-3D	-1.9%	+0.0%	-30.9%	-4.9%	-15.1%	-4.1%
HPSv3	-2.9%	-0.3%	-30.6%	-15.3%	+3.7%	+2.3%
EditReward ZS	—	—	0.793	0.679	0.676	0.661
EditReward FT	+13.3%	+7.0%	-1.4%	-1.5%	-13.2%	+2.0%
SR-PT+MLP	+34.0%	+22.7%	+4.7%	+3.7%	+28.3%	+18.9%
SR-PT+VarW	+35.2%	+22.5%	+5.7%	+5.2%	+28.1%	+18.9%
SR-PT+PR+VarW	+35.6%	+23.3%	+6.1%	+5.9%	+28.6%	+18.8%
SR-CA+PT+PR+VarW	+35.7%	+23.9%	+8.7%	+9.9%	+29.9%	+19.7%

and +27.5% on MagicBrush. This shows that explicit source-aware modeling preserves standard relative preference alignment.

For best-of- N ranking (Table 5), SourceReward improves top-1 selection across datasets. On Product Troubleshooting, SR-CA+PT+PR+VarW improves SuccessRate@1 by +26.5% over EditReward ZS, while improving NDCG@6 by +5.4%. The same trend holds on the public datasets, with SuccessRate@1 gains of +6.8% on EditReward and +28.0% on MagicBrush, indicating that the model can select usable edits from candidate sets.

The largest gains appear in accept/reject filtering (Table 6), where the model must decide whether a single edit is safe to use. SourceReward improves ROC-AUC by +35.7% and macro F1 by +23.9% on Product Troubleshooting, and improves ROC-AUC by +8.7% on EditReward and +29.9% on MagicBrush. EditReward FT does not consistently improve filtering performance on the public datasets despite fine-tuning, suggesting that adapting a reward model for relative edit preferences is not sufficient for threshold-based accept/reject

decisions. In contrast, SourceReward combines pointwise score prediction with pairwise supervision and explicitly models source adherence, better matching the filtering setting evaluated by ROC-AUC and macro F1.

To test alignment with human preferences, we additionally evaluate the trained reward model against the original TIGER-Lab annotator overall score on 1,577 non-tied test pairs (14.8% of pairs are excluded as integer-scale ties), reporting pairwise accuracy and image-level Spearman rank correlation (Table 7). This evaluation is particularly meaningful because these human scores were never seen during training: our model is trained exclusively on LLM-judge pseudo-labels generated via the cold-start pipeline, with no access to any human preference signal from the EditReward dataset. Despite this, SR-CA+PT+PR+VarW achieves a human pairwise accuracy of 0.819 and a Spearman rank correlation of 0.518 against held-out human judgments, indicating strong alignment between cold-start VLM supervision and genuine human preferences. These results suggest that an ensemble of VLM judges aggregated by Reciprocal Rank Fusion and weighted by inter-judge variance is a viable and scalable substitute for human annotation in cold-start regimes, particularly in domain-specific settings where collecting human preference data is expensive or infeasible.

4.4 Does Source Adherence Matter?

We isolate the role of source adherence by comparing SourceReward with EditReward ZS and EditReward FT, which condition on the source image, instruction, and edited image but do not model source adherence as a separate reward dimension. EditReward FT serves as a strong domain-adapted baseline, controlling for target-domain fine-tuning while retaining a reward formulation that primarily emphasizes instruction following and visual quality. On Product Troubleshooting, SourceReward improves over EditReward FT in both best-of- N selection and accept/reject filtering, with the gains most pronounced for filtering (+19.8% ROC-AUC and +15.8% F1, compared with +1.2% SuccessRate@1). This suggests that explicit source-adherence modeling helps select usable edits, but is especially useful for detecting plausible-looking edits that should be rejected due to source inconsistencies. The public-dataset results show the same trend beyond product imagery, with SourceReward improving ROC-AUC over EditReward ZS by +8.7% on EditReward and +29.9% on MagicBrush, while also improving SuccessRate@1 on both datasets.

Table 7: Trained reward model vs. held-out TIGER-Lab human preferences on 1,577 non-tied EditReward test pairs. PW Acc. is pairwise accuracy against the human *overall score* (instruction following + visual aesthetic quality); Spearman is image-level rank correlation.

Method	PW Acc. (human)	Spearman
SR-PT+VarW (ours)	0.812	0.498
SR-CA+PT+PR+VarW (ours)	0.819	0.518

4.5 Ablation: What Drives the Gains?

We ablate the components that distinguish SourceReward from standard reward modeling: variance-based uncertainty weighting, hybrid pointwise-pairwise training, and edit-centric cross-attention. Adding variance-based weighting during training in SR-PT+VarW improves

top-1 selection on product troubleshooting, increasing SR@1 from +23.1% to +27.3%, while preserving strong filtering performance. It also improves pairwise accuracy on both public datasets, from +1.3% to +1.9% on EditReward and from +24.4% to +25.5% on MagicBrush, suggesting that inter-judge disagreement provides a useful confidence signal under noisy cold-start supervision.

Adding pairwise ranking supervision in SR-PT+PR+VarW strengthens relative preference learning. It gives the strongest product troubleshooting pairwise gain of +4.9%, and also improves pairwise accuracy on EditReward and MagicBrush. This supports the role of pairwise supervision as a complement to pointwise score calibration.

Finally, adding edit-centric source/instruction cross-attention in SR-CA+PT+PR+VarW gives the most consistent filtering gains. The full model reaches +35.7% ROC-AUC and +23.9% F1 on product troubleshooting, improves to +8.7% ROC-AUC and +9.9% F1 on EditReward, and reaches +29.9% ROC-AUC and +19.7% F1 on MagicBrush. These trends suggest that uncertainty-aware weighting improves top-1 robustness, pairwise supervision improves relative preference learning, and edit-centric cross-attention provides the strongest gains for filtering by explicitly comparing edited regions against the source image and instruction.

Table 8: Estimated latency and cost for scoring $N = 6$ candidate edits across 100K source-instruction queries. VLM-based judging represents an inference-time VLM judge; EditReward FT and SR-CA+PT+PR+VarW are learned reward models. Reported latency measures scoring only and excludes non-scoring retry overhead.

Method	Scoring operations	Per-sample latency	Estimated compute	Estimated cost
VLM-based judging	600K VLM calls	$\sim 2,527$ ms	~ 300 API GPU-hours	$\sim \$10K$
EditReward FT	600K forward passes	~ 814 ms	~ 100 GPU-hours	$\sim \$250$
SR-CA+PT+PR+VarW	600K forward passes	~ 162 ms	~ 20 GPU-hours	$\sim \$50$

4.6 Efficiency and Inference-time Considerations

VLM-based judging provides a strong reference-grounded alternative for edit scoring and useful cold-start supervision, but is too expensive for high-volume inference-time reranking. Table 8 compares the cost of scoring $N = 6$ candidate edits across 100K source-instruction queries. Relative to VLM-based judging, SR-CA+PT+PR+VarW reduces latency from $\sim 2,527$ ms to ~ 162 ms, a $15.6\times$ speedup. reward models: relative to EditReward FT, SourceReward reduces reward-model scoring latency from ~ 814 ms to ~ 162 ms, a $5.0\times$ speedup. This efficiency comes from the scoring architecture: EditReward FT scores rubric dimensions through separate prompted forward passes, whereas SourceReward predicts all three reward heads in a single shared encoder pass. Thus, SourceReward preserves the benefits of reference-grounded scoring while substantially reducing the latency and cost of both VLM-based scoring and other finetuned reward-model scoring baselines.

5 Conclusion

We introduced SourceReward, a reward model for high-precision instruction-guided image editing that treats source adherence as a first-class objective alongside instruction relevance and visual integrity. By using edit-centric cross-attention to compare the edited image against

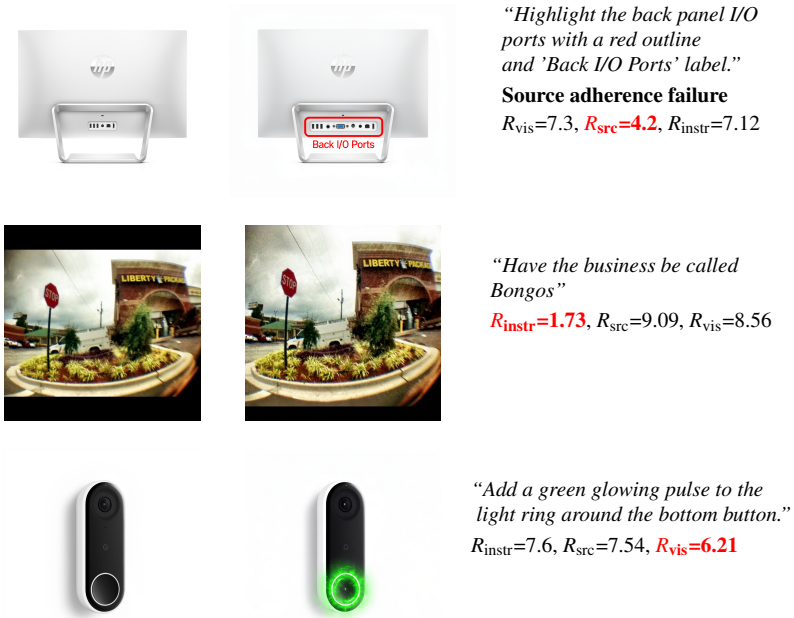


Figure 3: Failure modes captured by SourceReward. Each row shows source (left), edit (right), and per-dimension scores with the failing dimension in red. Row 1: Label added correctly but product structure distorted. Row 2: The business name remains unchanged despite the instruction to rename it to “Bongos.”. Row 3: The LED glow spreads unnaturally beyond the light ring, producing an unrealistic bloom artifact. This decomposition enables targeted diagnosis that a single composite score would obscure.

both the source image and the instruction, SourceReward targets a key weakness of existing editing reward models: plausible edits can follow the instruction while still altering source-specific structure that should remain unchanged. Across product troubleshooting, EditReward, and MagicBrush, SourceReward improves best-of- N selection and accept/reject filtering, with the strongest gains in the precision-critical product setting. It also reduces scoring latency by $15.6\times$ relative to VLM-based judging and $5.0\times$ relative to EditReward FT, making source-grounded reward modeling practical for high-volume reranking. Future work can extend SourceReward from candidate selection to reward-guided fine-tuning of image-editing models.

References

- [1] K. Black, M. Janner, Y. Du, I. Kostrikov, and S. Levine. Training diffusion models with reinforcement learning. In *ICLR*, 2024.
- [2] T. Brooks, A. Holynski, and A. A. Efros. InstructPix2Pix: Learning to follow image editing instructions. In *CVPR*, 2023.
- [3] J. Gu et al. Multi-reward as condition for diffusion models. In *ICLR*, 2024.

- [4] J. Hessel, A. Holtzman, M. Forbes, R. L. Bras, and Y. Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021.
- [5] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.
- [6] M. Kumar, S. Goyal, and D. Rosenberg. Better quantitative evaluation of text-to-image models. In *ACVC (Amazon internal)*, 2023.
- [7] P. Liu et al. PairJudge RM: Pairwise judge reward model for best-of-n sampling. *arXiv preprint*, 2025.
- [8] J. Luo et al. EditScore: An all-in-one reward model for scaling image editing with online reinforcement learning. *arXiv preprint*, 2025.
- [9] S. Park et al. GridAR: Progressive test-time compute scaling for autoregressive image generation. *arXiv preprint*, 2025.
- [10] Y. Qiao et al. TTGen: Test-time scaling with guidance for diffusion models. In *CVPR Workshop*, 2025.
- [11] Z. Qu et al. ADE-CoT: Adaptive chain-of-thought test-time scaling for image editing. *arXiv preprint*, 2026.
- [12] A. Sahay et al. Mirage: Metadata-guided image retrieval and answer generation for e-commerce troubleshooting. In *EACL*, 2026.
- [13] C. Snell, J. Lee, K. Xu, and A. Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint*, 2024.
- [14] X. Wang et al. ComplexBench-Edit: A multi-dimensional benchmark for fine-grained image editing. In *ACM MM*, 2025.
- [15] Y. Wang et al. SeedEdit 3.0: Image editing with causal diffusion and multi-granularity reward. *arXiv preprint*, 2025.
- [16] Z. Wang et al. ImAgent: Agentic test-time scaling for image generation. *arXiv preprint*, 2025.
- [17] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Shengming 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. *arXiv preprint*, 2025. arXiv:2508.02324.
- [18] H. Wu et al. Visual-RAG: A benchmark for text-to-image retrieval-augmented generation. *arXiv preprint*, 2025.
- [19] X. Wu et al. Human preference score: Better aligning text-to-image models with human preference. In *ICCV*, 2023.

- [20] Z. Wu et al. EditReward: A reward model for instruction-guided image editing with human-aligned preference learning. *arXiv preprint*, 2025.
- [21] J. Xu et al. ImageReward: Learning and evaluating human preferences for text-to-image generation. In *NeurIPS*, 2024.
- [22] Z. Xue et al. Enhanced multimodal rag-llm for accurate visual question answering. *arXiv preprint*, 2024.
- [23] X. Yang et al. Chart-MRAG Bench: Multimodal rag for chart understanding. *arXiv preprint*, 2025.
- [24] Q. Yu and M. Dreyer. Statistical analysis of human annotation noise in evaluation datasets. In *AMLC (Amazon internal)*, 2023.
- [25] Q. Yu, S. Ahn, J. Myung, Z. Lin, and H. Jin. Synthetic image statistic evaluation. In *ACVC (Amazon internal)*, 2023.
- [26] Z. Yuan et al. mKG-RAG: Multimodal knowledge graph retrieval-augmented generation. *arXiv preprint*, 2025.
- [27] K. Zhang, L. Mo, W. Chen, H. Sun, and Y. Su. MagicBrush: A manually annotated dataset for instruction-guided image editing. In *NeurIPS*, 2024.
- [28] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.