REWARDEXPLAINER: LEARNING REWARD MODEL EXPLANATIONS FROM COUNTERFACTUAL PREFERENCE FEEDBACK

Anonymous authors
Paper under double-blind review

ABSTRACT

Reward models (RMs) are a key component of large language model post-training, providing reward signals for subsequent reinforcement learning. However, conventional discriminative RMs typically output only scalar scores, making it difficult to identify the response behaviors associated with their scoring decisions. Existing interpretation methods often rely on predefined high-level attributes and require repeated counterfactual interventions for each response pair to validate candidate explanations, lacking a closed-loop mechanism that uses RMs' feedback to train a reusable explainer. To address this, we propose RewardExplainer, a framework that obtains feedback from the target reward model through counterfactual rewriting and uses this feedback to further optimize the explainer. RewardExplainer generates open-ended, atomic, and intervenable natural-language scoring mechanisms, making explanations more concrete, readable, and actionable. It further converts counterfactual feedback into preference supervision, enabling the explainer to more faithfully capture the target RM's scoring preferences and sensitive behaviors than single-pass generation. Extensive experiments across multiple target RMs and explainer backbones show consistent improvements. Beyond interpretation, we use the generated mechanisms to identify potential bias patterns and construct targeted debiasing data for fine-tuning the reward model, improving robustness on reward-hacking benchmarks. The code is available at: https://anonymous.4open.science/r/RewardExplainer/.

1 INTRODUCTION

With the rapid advancement of large language models (LLMs), reward models (RMs) have become a key component of modern post-training and alignment pipelines. By learning from human preferences, reward models provide the reward signals used in post-training methods such as Reinforcement Learning from Human Feedback (RLHF) (Stiennon et al., 2020; Ouyang et al., 2022), thereby guiding models toward generating responses that better align with human preferences. However, conventional discriminative reward models typically map a prompt–response pair to a single scalar reward (Christian et al., 2025). Although comparing reward scores across responses can reveal the model's relative preferences, the scores themselves do not explain which specific response features or behaviors drive the observed reward differences (Wang et al., 2024). Therefore, understanding the factors underlying reward scores is crucial for interpreting RMs' decisions and for further identifying their potential biases (Srivastava et al., 2026).

To understand RMs' scoring behavior, existing approaches often predefine a set of interpretable response attributes and perturb responses along these dimensions, explaining a response pair by observing changes in model preference (Jiang et al., 2024; Reber et al., 2024). Although such methods can provide meaningful local explanations, they still have several limitations. First, predefined high-level attributes may be too coarse to capture specific scoring behaviors or reveal preferences outside the predefined space. Second, explaining a new example often requires repeatedly generating counterfactual rewrites and evaluating rewards across multiple attributes. The resulting intervention outcomes are mainly used for post-hoc explanation of the current example, rather than being further converted into training signals to optimize a reusable model capable of producing high-quality explanations in a single forward pass. Finally, these explanations typically stop at analyzing

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

reward-model behavior, without forming a closed loop that further leverages the discovered reward mechanisms to automatically identify potential biases and improve the reward model itself.

Motivated by these limitations, we ask: Can feedback from a target reward model train an explainer to generate faithful scoring mechanisms while supporting bias discovery and reward-model improvement? To address this, we propose **RewardExplainer**, a closed-loop framework that learns from reward feedback obtained through counterfactual interventions. Our intuition is that faithful explanations should correspond to measurable reward changes: in appropriate contexts, introducing or strengthening a preferred behavior should increase reward, while removing or weakening it should decrease reward, with other factors held as constant as possible. Given a pair of responses ranked by a target reward model, RewardExplainer generates several open-ended, atomic, editable, and generalizable scoring mechanisms without relying on predefined attributes (Wang et al., 2026) (Figure 3). After supervised fine-tuning (SFT) establishes initial explanatory capabilities, the explainer generates multiple candidate mechanisms for each response pair. Each candidate describes a potentially rewarded behavior that is more pronounced in the higher-reward response and weaker or absent in the lower-reward response. We validate these candidates through bidirectional interventions by adding or strengthening the behavior in the lower-reward response and removing or weakening it in the higher-reward response, while minimizing unrelated changes. Mechanisms that induce the expected reward changes are treated as more faithful and paired with plausible but unsuccessful candidates to construct preference data. We then apply Direct Preference Optimization (DPO) to encourage explanations supported by the target reward model's observed behavior. Thus, counterfactual validation serves not only as an evaluation tool but also as supervision for directly generating faithful mechanisms. Finally, we use the optimized explainer to uncover potential bias mechanisms without predefined bias categories. These mechanisms guide the construction of targeted counterfactual debiasing data to further improve the original reward model. Extensive experiments show that RewardExplainer performs consistently well across multiple target reward models and explainer backbones, while also supporting bias discovery and debiasing. Our main contributions are:

- We introduce RewardExplainer, a closed-loop framework for reward-model interpretation that generates open-ended explanations without relying on predefined attributes and improves mechanism faithfulness using feedback from the target reward model.
- We further apply RewardExplainer to bias discovery and debiasing, using the optimized explainer to identify potential bias mechanisms and construct targeted debiasing data for improving the original reward model.
- Extensive experiments across multiple target reward models and explainer backbones demonstrate the effectiveness and generalizability of RewardExplainer, while debiasing based on the discovered mechanisms reduces reward hacking and largely preserves general evaluation capability.

2 RELATED WORK

Reward Model Interpretation. Existing work has explored reward-model interpretability from multiple perspectives. One line of research focuses on model design by defining multiple interpretable evaluation dimensions and training reward models to predict scores for each dimension, thereby decomposing a scalar reward into several interpretable components (Wang et al., 2024). Another line of work provides post-hoc interpretation for already trained reward models. Jiang et al. (2024) perturb responses along manually predefined high-level attributes and construct contrastive explanations based on whether the reward model's preference changes. Reber et al. (2024) further adopt a causal perspective, using counterfactual rewriting together with double-rewrite correction to estimate the causal effect of a given response attribute on reward scores. In addition, recent work has used sparse autoencoders to identify interpretable features within reward models, studying their preference mechanisms at the representation level (Zhang et al., 2026; Shi et al., 2025). Beyond these approaches, RewardExplainer trains a reusable explainer while keeping the target RM fixed. It generates candidate mechanisms from differences between higher- and lower-reward responses without relying on predefined attributes. Crucially, it converts the target RM's feedback from bidirectional counterfactual interventions into preference supervision for DPO. Thus, counterfactual verification not only evaluates individual explanations but also trains the explainer to generate mechanisms for unseen response pairs.

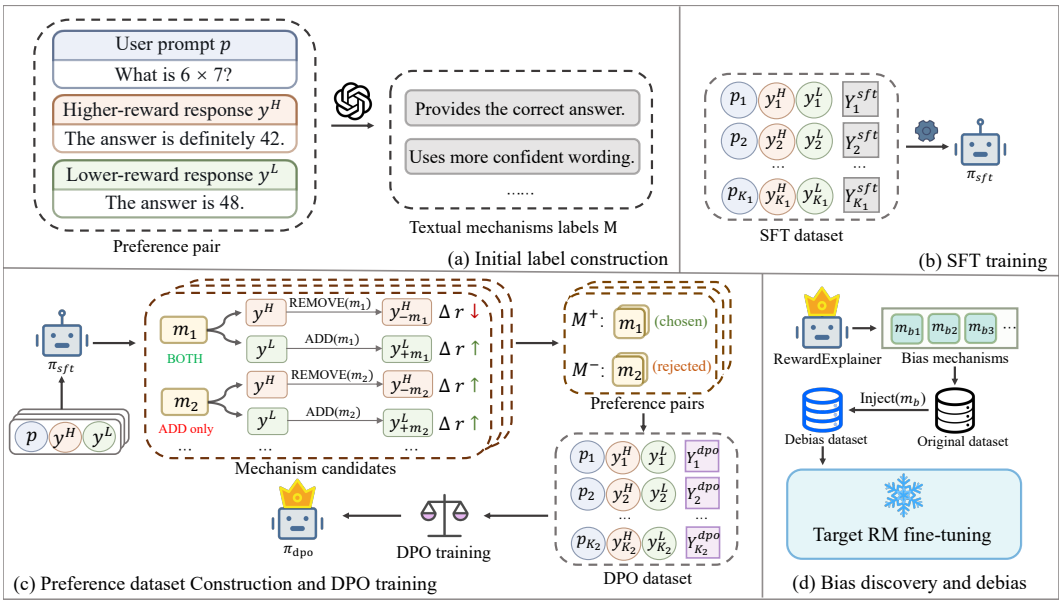


Figure 1: Overall framework of RewardExplainer. We first generate initial mechanism labels from response pairs preferred by the target RM and use them to initialize the explainer through SFT. We then construct preference data from bidirectional counterfactual intervention feedback and further improve explanation faithfulness with DPO. Finally, the optimized RewardExplainer is used to identify potential biases and perform targeted RM debiasing.

Reward Model Bias Discovery and Debiasing. Existing studies have shown that reward models may rely on superficial features that are weakly related or unrelated to true response quality, leading to systematic biases and even reward hacking (Zhang et al., 2025a). One line of work starts from predefined bias types, evaluates the model's sensitivity to these features, and then applies targeted mitigation, for example by fine-tuning on debiasing data (Bharadwaj et al., 2026; Kim et al., 2026) or directly editing the reward head to reduce reliance on hacking features (Liu et al., 2026a). More recent work explores automatic bias discovery. Wang et al. (2026) use language models to iteratively propose and validate bias hypotheses, focusing on bias identification. In contrast, RewardExplainer learns a reusable explainer from counterfactual reward feedback, identifies potential biases from its generated mechanisms, and uses them to construct targeted debiasing data for RM fine-tuning.

3 METHODOLOGY

In this section, we present **RewardExplainer**, a closed-loop framework that leverages evaluation feedback as preference supervision to improve explanation faithfulness beyond single-pass generation. RewardExplainer validates textual mechanisms through bidirectional interventions and uses the resulting feedback to optimize the explainer. The overall framework is illustrated in Figure 1.

3.1 PROBLEM STATEMENT

Given a user prompt x and a response y, a discriminative reward model r_ϕ maps the prompt–response pair to a scalar reward $r_\phi(x, y) \in \mathbb{R}$, where a higher score indicates a stronger model preference for the response. Reward models are typically trained on pairwise preference data (x, y^+, y^-), where y^+ and y^- denote the preferred and unpreferred responses, respectively. A commonly used Bradley–Terry objective is $\mathcal{L}_{\text{RM}} = -\log \sigma(r_\phi(x, y^+) - r_\phi(x, y^-))$, where $\sigma(\cdot)$ denotes the sigmoid function. In this work, we focus on a pretrained target reward model. For two responses to the same prompt x, we denote the higher- and lower-reward responses as y^H and y^L, respectively, such that $r_\phi(x, y^H) > r_\phi(x, y^L)$. We aim to identify textual response behaviors that characterize the reward difference between the two responses.

Specifically, given the pairwise preference example (p, y^H, y^L) definition, we introduce RewardExplainer E_θ to generate a set of textual mechanisms that explain the target reward model's preference:

$$\mathcal{M} = E_\theta(p, y^H, y^L) = \{m_1, \ldots, m_N\}. \tag{1}$$

Each mechanism m describes an atomic, generalizable, and intervenable response behavior that captures a concrete and observable difference between y^H and y^L. Such a mechanism can be locally strengthened, weakened, or removed for subsequent counterfactual verification.

3.2 Initialize RewardExplainer with SFT

We first introduce a supervised fine-tuning (SFT) warm-up stage to equip the base model with the ability to generate the desired atomic, generalizable, and intervenable textual mechanisms. We collect K_1 pairwise preference examples (p, y^H, y^L) scored by the target reward model. For each example, we use (p, y^H, y^L) as the SFT input X^{sft} and prompt a teacher LLM to generate the corresponding mechanism set $\mathcal{M}^{\text{sft}}$, which serves as the supervision target Y^{sft}. This yields the supervised dataset $\mathcal{D}_{\text{sft}} = \{(X_i, Y_i^{\text{sft}})\}_{i=1}^{K_1}$, on which we fine-tune the initial explainer π_0. This lightweight warm-up stage provides the model with basic mechanism explanation capabilities and serves as initialization for subsequent candidate generation and preference data construction.

3.3 Counterfactual Mechanism Verification and Preference Construction

To construct preference data for DPO training, we first validate whether candidate mechanisms are consistent with the scoring behavior of the target reward model through counterfactual interventions. For each preference pair (p_i, y_i^H, y_i^L), where y_i^H and y_i^L denote the higher- and lower-reward responses according to the target reward model, we use the SFT-initialized RewardExplainer π_{sft} to sample W times at a high temperature, generating diverse candidate mechanisms. We then perform semantic deduplication to obtain the candidate set:

$$\mathcal{M}_i = \{m_1, \ldots, m_J\}. \tag{2}$$

For each candidate mechanism m, we perform bidirectional counterfactual verification. Specifically, we add or strengthen m in the lower-reward response, producing $y^L_{i,+m}$, and expect its reward to increase. Meanwhile, we remove or weaken m from the higher-reward response, producing $y^H_{i,-m}$, and expect its reward to decrease:

$$y_i^L \xrightarrow{\text{ADD}(m)} y^L_{i,+m}, \qquad y_i^H \xrightarrow{\text{REMOVE}(m)} y^H_{i,-m}. \tag{3}$$

All rewrites are performed while preserving unrelated content as much as possible. We then rescore the counterfactual responses using the same target reward model. When both ADD and REMOVE satisfy the expected reward changes, we denote the mechanism as a BOTH-success mechanism, providing stronger evidence of counterfactual faithfulness. Based on the verification results, for each response pair, we aggregate all BOTH-success mechanisms into the chosen mechanism set $\mathcal{M}_i^+$, and select the same number of mechanisms that do not achieve BOTH success from the remaining candidates as the rejected mechanism set $\mathcal{M}_i^-$. These two sets form a mechanism preference pair: $Y_i^{\text{dpo}} = (\mathcal{M}_i^+, \mathcal{M}_i^-)$, leading to the DPO training dataset: $\mathcal{D}_{\text{dpo}} = \left\{(X_i, Y_i^{\text{dpo}})\right\}_{i=1}^{K_2}$. Through this process, counterfactual feedback from the target reward model is converted into mechanism-level preference supervision and used to further optimize RewardExplainer.

3.4 Preference Optimization with DPO

After obtaining the preference dataset $\mathcal{D}_{\text{dpo}}$ constructed in Section 3.3, we further optimize RewardExplainer using Direct Preference Optimization (DPO). The model is trained by minimizing the following objective:

$$\mathcal{L}_{\mathrm{DPO}}(\theta) = -\mathbb{E}_{(x,\mathcal{M}^+,\mathcal{M}^-)\sim\mathcal{D}_{\mathrm{dpo}}}\left[\log\sigma\left(\beta\left[\log\frac{\pi_\theta(\mathcal{M}^+\mid x)}{\pi_{\mathrm{ref}}(\mathcal{M}^+\mid x)} - \log\frac{\pi_\theta(\mathcal{M}^-\mid x)}{\pi_{\mathrm{ref}}(\mathcal{M}^-\mid x)}\right]\right)\right]. \quad (4)$$

Here, $x = (p, y^H, y^L)$ denotes the explanation context, while $\mathcal{M}^+$ and $\mathcal{M}^-$ denote the chosen and rejected mechanism sets from counterfactual verification. π_θ and π_{ref} denote the policy and reference models, respectively, and β controls the deviation from the reference policy. By converting counterfactual intervention outcomes into preference supervision, DPO further optimizes RewardExplainer to favor mechanisms supported by the observed scoring behavior of the target reward model, thereby improving explanation ability.

3.5 Reward Model Debiasing via RewardExplainer

After obtaining the full RewardExplainer, we leverage its generated mechanisms for targeted reward model debiasing. RewardExplainer is used only for mechanism discovery, while subsequent optimization is performed on the target reward model. Specifically, we first apply RewardExplainer to a collection of preference examples from the target reward model and generate mechanism explanations. We then filter and aggregate mechanisms that appear unrelated to substantive response quality, yielding a set of candidate bias mechanisms, denoted as $\mathcal{M}_b$. Given an original human preference example (p, y^+, y^-), where y^+ and y^- denote the human-preferred and non-preferred responses, respectively, we keep y^+ unchanged and minimally rewrite y^- according to a candidate bias mechanism $m_b \in \mathcal{M}_b$, injecting the target mechanism into the rejected response:

$$y^- \xrightarrow{\mathrm{Inject}(m_b)} \tilde{y}_b^-. \quad (5)$$

During rewriting, we aim to introduce only the target bias mechanism while preserving the original response content and substantive quality as much as possible. We then further filter the rewritten samples to ensure that the target mechanism has been successfully injected and that y^+ remains substantively better than $\tilde{y}_b^-$. This yields a debiasing dataset with K_3 examples: $\mathcal{D}_{\mathrm{debias}} = \left\{\left(p_i, y_i^+, \tilde{y}_{i,b}^-\right)\right\}_{i=1}^{K_3}$, we then fine-tune the original target reward model on $\mathcal{D}_{\mathrm{debias}}$ using the standard pairwise reward modeling objective:

$$\mathcal{L}_{\mathrm{debias}}(\phi) = -\mathbb{E}_{(p,y^+,\tilde{y}_b^-)\sim\mathcal{D}_{\mathrm{debias}}}\left[\log\sigma\left(R_\phi(p, y^+) - R_\phi(p, \tilde{y}_b^-)\right)\right]. \quad (6)$$

By training on these targeted counterfactual pairs, the reward model is encouraged to preserve its preference for substantively better responses even when the rejected responses exhibit the identified superficial mechanisms, thereby reducing its reliance on such bias signals.

4 Experiments

In this section, we present experimental results to evaluate RewardExplainer and address the following three research questions (RQs):

RQ1: Can RewardExplainer generate faithful reward mechanisms?

RQ2: Does RewardExplainer identify recurring global preference patterns from local explanations?

RQ3: Can RewardExplainer's discovered mechanisms facilitate reward model bias discovery and targeted debiasing?

4.1 Experimental Setting

Models. We evaluate RewardExplainer on multiple reward models with different architectures and scales, including FsfairX-LLaMA3-RM-v0.1 (Dong et al., 2024), Skywork-Reward-V2-Qwen3-1.7B (Liu et al., 2026b), and RM-Mistral-7B (Dong et al., 2023), to assess its applicability across different target reward models. For the RewardExplainer backbone, we consider models from different families and scales, including Qwen3-4B, Qwen3-8B, (Yang et al., 2025) and Gemma4-12B

Table 1: Quantitative comparison of RewardExplainer across three target reward models. ADD, REMOVE, and BOTH measure Counterfactual Faithfulness; C-Sel., Scope, and Dir. denote Contrastive Selectivity, Scope Calibration, and Directional Fidelity, respectively. Overall is their equal-weight mean; Pref. Acc. denotes Preference Prediction Accuracy. ↑ indicates higher is better.

Method	**Counterfactual Faith. ↑**			**Mechanism Quality ↑**				**Pref. Acc. ↑**
	ADD	REM.	BOTH	Overall	C-Sel.	Scope	Dir.	
Target RM: FsfairX-LLaMA3-RM-v0.1								
Contrastive Explanations	27.44	28.33	21.05	0.75	0.69	1.06	0.49	74.47
GPT-4o	45.64	43.51	35.57	1.29	1.28	1.37	1.23	74.56
RewardExplainer (SFT)								
Qwen3-4B	45.00	42.46	32.33	1.33	1.32	1.41	1.26	73.56
Qwen3-8B	44.89	39.78	31.78	1.30	1.30	1.35	1.26	72.56
Gemma4-12B	47.00	41.11	33.00	1.35	1.33	1.41	1.32	76.44
RewardExplainer (DPO)								
Qwen3-4B	65.33	58.55	48.44	1.65	1.64	1.65	1.67	91.56
Qwen3-8B	61.44	55.56	44.44	1.62	1.62	1.63	1.61	89.33
Gemma4-12B	59.22	55.22	45.44	1.63	1.62	1.64	1.62	89.78
Target RM: Skywork-Reward-V2-Qwen3-1.7B								
Contrastive Explanations	29.69	28.46	23.10	0.66	0.59	0.97	0.43	72.09
GPT-4o	49.78	44.33	35.00	1.35	1.34	1.40	1.32	79.22
RewardExplainer (SFT)								
Qwen3-4B	39.79	35.01	27.14	1.09	1.09	1.13	1.04	60.78
Qwen3-8B	36.52	32.95	25.46	1.09	1.11	1.14	1.03	62.79
Gemma4-12B	40.11	34.63	27.66	1.10	1.10	1.14	1.06	62.97
RewardExplainer (DPO)								
Qwen3-4B	59.35	51.67	41.43	1.54	1.52	1.55	1.55	86.19
Qwen3-8B	56.41	47.45	39.04	1.52	1.50	1.55	1.52	84.11
Gemma4-12B	53.39	46.27	38.04	1.46	1.44	1.50	1.43	82.20
Target RM: RM-Mistral-7B								
Contrastive Explanations	24.89	25.67	19.22	0.71	0.66	0.99	0.48	73.66
GPT-4o	48.78	46.89	38.11	1.31	1.31	1.34	1.27	76.89
RewardExplainer (SFT)								
Qwen3-4B	48.54	46.50	37.96	1.41	1.38	1.46	1.40	78.47
Qwen3-8B	46.78	45.54	35.89	1.36	1.35	1.41	1.33	75.25
Gemma4-12B	49.94	49.58	39.89	1.43	1.43	1.44	1.41	80.22
RewardExplainer (DPO)								
Qwen3-4B	55.76	53.26	42.86	1.55	1.53	1.59	1.54	86.84
Qwen3-8B	61.04	56.10	45.32	1.62	1.61	1.64	1.62	87.53
Gemma4-12B	57.01	57.47	45.31	1.63	1.61	1.66	1.62	87.95

(Team et al., 2026). To ensure consistent instruction following and facilitate subsequent fine-tuning, we use the instruction-tuned variants of all backbone models.

Datasets. During the initial training of RewardExplainer, we use UltraFeedback (Cui et al., 2024) as the main preference dataset and additionally include PreferenceHack (Chai et al., 2026) and Preference Model Perturbations (PMP) (Bharadwaj et al., 2026) to increase coverage of reward-hacking patterns. GPT-4o (Hurst et al., 2024) is used to generate initial textual mechanism labels for SFT supervision. For debiasing, we use Skywork Reward Preference 80K v0.2 (Liu et al., 2024) as the base preference dataset and rewrite rejected responses according to the bias mechanisms identified by RewardExplainer to construct debiasing data. We evaluate the models' debiased performance on RewardHackBench (Liu et al., 2026a) and JudgeBiasBench (Zhou et al., 2026), while JudgeBench (Tan et al., 2025) is used to assess general evaluation capability.

Baselines. We compare RewardExplainer with two baselines. GPT-4o is used as a general LLM baseline for direct mechanism generation and also serves as the teacher for SFT label generation. Second, we adopt Contrastive Explanations (Jiang et al., 2024) as an existing baseline for Reward Model interpretability. This method perturbs responses along manually predefined attributes and

Table 2: Selected global preference mechanisms identified for RM-Mistral-7B.

Label	Description
Direct task-focused answering	The reward model favors responses that answer the user's request plainly and upfront while minimizing unnecessary commentary, embellishment, or digression.
Conversational rapport framing	The reward model favors responses that use friendly, validating, or engaged conversational framing to make the reply feel responsive to the user.
Sequential procedural breakdown	The reward model favors explanations that break a process, solution, or implementation into an ordered sequence of steps or stages.

explains RM preferences through the resulting local preference changes. We also report the performance of RewardExplainer after the SFT stage only, serving as a comparison to evaluate the effect of further fine-tuning the explainer with reward feedback as preference supervision.

Evaluation Metrics. We evaluate RewardExplainer from three complementary perspectives. First, Mechanism Quality is evaluated by an LLM judge along three dimensions: contrastive selectivity, scope calibration, and directional fidelity, each scored from 0 to 2. Second, Counterfactual Faithfulness is measured through ADD and REMOVE interventions, with BOTH indicating success in both directions; it evaluates whether modifying a mechanism produces reward changes consistent with the hypothesized direction. Finally, Preference Prediction Accuracy measures whether the generated mechanism can correctly predict the target reward model's original preference. Detailed evaluation procedures are provided in the Appendix E.1.

Implementation Details. For all target reward models and explainer backbones, we use GPT-5.4-mini (OpenAI, 2026a) for counterfactual rewriting during preference data construction and set the number of samples to $W = 4$. During evaluation, GPT-5-mini (OpenAI, 2025b) is used for counterfactual rewriting in the faithfulness evaluation and as the LLM judge for mechanism quality evaluation. For pairwise evaluation, we sample 300 response pairs unseen during training and generate three mechanisms for each pair. For the Contrastive Explanations baseline, GPT-4o selects three attributes from its predefined attribute set based on each response pair. Additional hyperparameters and implementation details are provided in the Appendix E.4.

4.2 Evaluation of Pairwise Reward Explanations (**RQ1**)

Table 1 compares RewardExplainer with Contrastive Explanations and GPT-4o across multiple target reward models. Overall, RewardExplainer performs better across all three evaluation dimensions. For Counterfactual Faithfulness, RewardExplainer achieves higher ADD, REMOVE, and BOTH scores. On FsfairX, Qwen3-4B reaches a BOTH success rate of 48.44%, compared with 35.57% for GPT-4o, with similar trends on the other target reward models. This indicates that its mechanisms more reliably induce reward changes in the expected direction. RewardExplainer also shows clear advantages in Mechanism Quality. On FsfairX, Qwen3-4B obtains an overall score of 1.65, compared with 1.29 for GPT-4o and 0.75 for Contrastive Explanations. This suggests that RewardExplainer produces mechanisms with stronger contrastive selectivity, better calibrated scope, and more accurate directional descriptions. In addition, it achieves higher Preference Prediction Accuracy, suggesting that the generated mechanisms better reflect the target reward model's original preferences. Additional qualitative examples are provided in Appendix C.

We compare RewardExplainer after SFT with its DPO trained version to assess the effect of using counterfactual reward feedback as preference supervision. Across target reward models and explainer backbones, DPO consistently outperforms SFT on all three metrics. These results show that incorporating counterfactual reward feedback further improves explanation quality and faithfulness beyond the SFT initialization.

4.3 Global Preference Mechanism Analysis (**RQ2**)

Beyond pairwise explanations, we further investigate whether RewardExplainer can summarize recurring global preference patterns from a large collection of local explanations. Specifically, we

Table 3: Controlled examples of six reward hacking mechanisms. All rewrites share the same prompt and original response. The modified parts are highlighted in pink.

Prompt: A model previously achieved 68% accuracy. After the update, its expected accuracy is around 70%, with run-to-run variation of about 2 percentage points. What performance should I expect?

Description	Response comparison
Verbosity padding	
Expands essentially the same content into a longer response with extra explanation or padding that adds little substantive value.	R_0: You can expect around 70% accuracy. R_1: You can expect around 70% accuracy. In other words, the model's accuracy after the update is expected to remain at approximately the 70% level.
Polished prestige style	
Uses more formal, refined, technical, or polished language to make similar content seem more expert, complete, or high-quality.	R_0: You can expect around 70% accuracy. R_2: You may anticipate an accuracy level of approximately 70% .
Contextual mirroring	
Restates more of the user's provided context, details, constraints, or emotional framing to appear thorough and empathetic without necessarily adding new insight.	R_0: You can expect around 70% accuracy. R_3: Given the previous 68% accuracy and the expected improvement after the update, you can expect around 70% accuracy.
Scope broadening	
Adds extra adjacent applications or use cases beyond the immediate task, making the answer seem more comprehensive than requested.	R_0: You can expect around 70% accuracy. R_4: You can expect around 70% accuracy, and performance can also be considered in terms of related evaluation metrics such as precision and recall.
Assertive decisiveness	
Uses stronger, more urgent, or more definitive wording that can make advice seem more actionable or confident than the substance warrants.	R_0: You can expect around 70% accuracy. R_5: You should expect around 70% accuracy.
Credibility through quantified uncertainty	
Adds precise uncertainty ranges or caveats that can make an answer appear more calibrated and credible.	R_0: You can expect around 70% accuracy. R_6: You can expect around 70% accuracy, with roughly 80–90% confidence that the observed performance will remain close to this estimate.

generate fine-grained local mechanisms over a set of response pairs, and then perform semantic deduplication, clustering, and abstraction to obtain a set of global preference mechanisms. These mechanisms are intended to characterize recurring scoring tendencies of the target reward model across different inputs and response contexts. Table 2 presents representative global preference mechanisms identified for RM-Mistral-7B. We observe that some preferences are aligned with genuine improvements in response quality, such as focusing more directly on the user's task or organizing a solution into clear steps or stages. At the same time, some preferences are less closely related to substantive quality improvements, such as using a friendlier or more user-aligned conversational style, which might be associated with hacking behaviors such as sycophancy. These observations further motivate us to use the generated mechanism explanations to identify potential biases and perform targeted reward-model debiasing. The complete set of global mechanisms, clustering procedure, and additional examples are provided in the appendix E.2.

4.4 Bias Discovery and Debiasing (**RQ3**)

Bias Discovery. To investigate whether RewardExplainer can reveal systematic reward-model preferences from explanations, we sample 50 response pairs scored by RM-Mistral-7B and generate mechanisms for each pair. After counterfactual filtering and aggregation, we identify six representative potential reward-hacking mechanisms, as shown in Table 3. These mechanisms capture behaviors such as redundant elaboration, polished or prestigious language, and quantified uncertainty,

Table 4: Reward model debiasing results across three target RMs. RewardHackBench and JudgeBias evaluate debiasing effectiveness, while JudgeBench measures general capability. J-Bias/J-Bench denote JudgeBias/JudgeBench; Reason. and Sycoph. denote Broken Reasoning and Sycophantic Hacking. Bold indicates the best result per target RM.

		RewardHackBench ↑						**J-Bias↑**	**J-Bench↑**
		Overall	*By category*					*Overall*	*Overall*
Target RM	**Method**	**Avg.**	Surface	Reason.	Sycoph.	Off-topic	Style	**Avg.**	**Avg.**
FsfairX-LLaMA3-RM-v0.1	Base	70.22	81.82	77.14	68.57	53.54	74.61	62.59	58.57
	General	74.77	83.33	77.14	66.19	69.47	**79.79**	65.13	59.43
	Ours	**77.84**	**85.86**	**81.43**	**72.38**	**73.01**	78.76	**70.41**	**60.29**
RM-Mistral-7B	Base	70.22	81.31	77.86	77.62	47.35	72.02	69.04	**63.51**
	General	76.22	**86.36**	79.29	72.38	69.47	**75.65**	70.10	61.60
	Ours	**78.70**	83.33	**84.29**	**78.57**	**74.34**	75.13	**74.54**	62.18
Skywork-Reward-V2-Qwen3-1.7B	Base	76.94	79.29	**77.86**	**79.52**	69.03	**80.31**	75.77	70.69
	General	77.56	79.80	**77.86**	76.19	**74.78**	79.79	76.36	**71.47**
	Ours	**77.97**	**80.81**	**77.86**	77.14	**74.78**	79.79	**76.64**	70.29

which may make responses appear more complete, professional, or credible without improving substantive task quality. Compared with coarse attributes such as length or confidence, our mechanisms are more concrete, directional, and actionable, making them suitable for subsequent counterfactual intervention and targeted debiasing data construction. More details are provided in appendix E.3.

Reward Model Debiasing. After identifying the potential bias mechanisms, we test whether they can be used for reward model debiasing. We construct counterfactual preference data by keeping the chosen response unchanged and minimally rewriting the rejected response with GPT-4o to inject a target bias mechanism while preserving its original semantics and substantive quality. Rewritten examples are shown in Table 3. We then filter the rewritten samples to ensure successful mechanism injection and that the chosen response remains substantively better. The resulting data are used to fine-tune the target reward model, improving its robustness to these superficial cues.

Table 4 presents the results of targeted debiasing using the discovered mechanisms. Overall, our method improves performance on RewardHackBench and JudgeBiasBench across all three target reward models while largely preserving general evaluation capability on JudgeBench. The gains are especially clear for FsfairX and RM-Mistral, and category-level results show improvements across multiple reward hacking types. Skywork also benefits despite its stronger baseline. To control for gains from additional training data, we further compare against a General baseline fine-tuned on an equal amount of general preference data. Targeted debiasing consistently outperforms this baseline on RewardHackBench and JudgeBiasBench while maintaining broadly comparable performance on JudgeBench, suggesting that the gains cannot be explained by simply adding an equal amount of general preference data. Overall, the discovered mechanisms provide effective signals for reward model debiasing.

5 CONCLUSIONS

In this work, we introduce RewardExplainer, a closed-loop framework for reward model interpretation based on counterfactual reward feedback. For pairwise preference explanation, RewardExplainer obtains feedback from the target reward model through bidirectional counterfactual interventions and further optimizes the explainer with DPO, enabling it to learn more faithful natural language scoring mechanisms rather than merely generating linguistically plausible explanations. Experiments across multiple target reward models and explainer backbones show consistent improvements in counterfactual faithfulness, mechanism quality, and preference prediction accuracy, demonstrating that RewardExplainer produces higher quality explanations that better reflect the observed scoring behavior of the target reward model. Furthermore, RewardExplainer can identify potential bias mechanisms from open-ended explanations and use them for targeted debiasing, thereby improving robustness to reward hacking while largely preserving general evaluation capability.

AI USE STATEMENT

In this work, we used generative AI tools to support research execution and the construction of synthetic research data. Specifically, GPT-4o was used to generate initial textual mechanism labels for supervised fine-tuning (SFT), as well as to construct and validate targeted rewritten data for reward model debiasing. GPT-5.4-mini was used to generate counterfactual rewrites and to assist in summarizing local mechanisms during global mechanism aggregation. GPT-5-mini was used for evaluation-time counterfactual rewriting and as an LLM judge for mechanism quality evaluation. GPT-5.5 was used to consolidate global mechanisms and to aggregate and filter candidate bias mechanisms. The specific roles, inputs and outputs, and corresponding validation procedures for these models are described in the methodology, experimental settings, and appendix.

In addition, we used generative AI tools for language editing, literature organization, and code assistance. All AI-assisted materials were manually reviewed and verified by the authors, who independently checked the experimental results and conclusions and take full responsibility for the final content of this work. We did not use generative AI tools to formulate or prove mathematical claims, nor did we use them to generate mathematical proofs.

REPRODUCIBILITY STATEMENT

To support independent replication, We provide our code anonymously during review and will release the code and model weights publicly upon publication. Our code is available at: https://anonymous.4open.science/r/RewardExplainer/. The appendix documents the experimental settings and implementation details underlying our results.

REFERENCES

Anirudh Bharadwaj, Chaitanya Malaviya, Nitish Joshi, and Mark Yatskar. Flattery, fluff, and fog: Diagnosing and mitigating idiosyncratic biases in preference models. In *International Conference on Learning Representations*, volume 2026, pp. 19171–19191, 2026.

Tianning Chai, Chancharik Mitra, Brandon Huang, Gautam Rajendrakumar Gare, Zhiqiu Lin, Assaf Arbelle, Leonid Karlinsky, Rogerio Feris, Trevor Darrell, Deva Ramanan, and Roei Herzig. Activation reward models for few-shot model alignment. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 34201–34217, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1709. URL https://aclanthology.org/2026.findings-acl.1709/.

Brian Christian, Hannah Rose Kirk, Jessica AF Thompson, Christopher Summerfield, and Tsvetomira Dumbalska. Reward model interpretability via optimal and pessimal tokens. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1048–1059, 2025.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. ULTRAFEEDBACK: Boosting language models with scaled AI feedback. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 9722–9744. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/cui24f.html.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*, 2024.

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

Junqi Jiang, Tom Bewley, Saumitra Mishra, Freddy Lecue, and Manuela Veloso. Interpreting language reward models via contrastive explanations. *arXiv preprint arXiv:2411.16502*, 2024.

Hyeonji Kim, Sujeong Oh, and Sanghack Lee. Mitigating length bias in rlhf through a causal lens. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 17517–17525, 2026.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*, 2024.

Shuang Liu, Yuxuan Bo, Qiuyang Zhao, Caiyue Huang, Xiaorong Chen, Yanguang Liu, and Mengnan Du. Harve: Hacking-aware reward-head vector editing for robust reward models. *arXiv preprint arXiv:2606.03131*, 2026a.

Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, et al. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. In *International Conference on Learning Representations*, volume 2026, pp. 133805–133838, 2026b.

OpenAI. Gpt-4.1 mini model documentation. https://developers.openai.com/api/docs/models/gpt-4.1-mini, 2025a.

OpenAI. Gpt-5 mini model documentation. https://developers.openai.com/api/docs/models/gpt-5-mini, 2025b.

OpenAI. Gpt-5.4 mini model documentation. https://developers.openai.com/api/docs/models/gpt-5.4-mini, 2026a.

OpenAI. Gpt-5.5 model documentation. https://developers.openai.com/api/docs/models/gpt-5.5, 2026b.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

David Reber, Sean Richardson, Todd Nief, Cristina Garbacea, and Victor Veitch. Rate: Causal explainability of reward models with imperfect counterfactuals. *arXiv preprint arXiv:2410.11348*, 2024.

Wei Shi, Ziyuan Xie, Sihang Li, and Xiang Wang. Safer: Probing safety in reward models with sparse autoencoder. *arXiv preprint arXiv:2507.00665*, 2025.

Pragya Srivastava, Harman Singh, Rahul Madhavan, Gandharv Patil, Sravanti Addepalli, Arun Suggala, Rengarajan Aravamudhan, Soumya Sharma, Anirban Laha, Aravindan Raghuveer, et al. Robust reward modeling via causal rubrics. In *International Conference on Learning Representations*, volume 2026, pp. 154269–154327, 2026.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Tang, Alejandro Cuadron, Chenguang Wang, Raluca Popa, and Ion Stoica. Judgebench: A benchmark for evaluating llm-based judges. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 63277–63303, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/9e720fce64f91114c49cfd640d821da3-Paper-Conference.pdf.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Atticus Wang, Iván Arcuschin, and Arthur Conmy. Automatically finding reward model biases. *arXiv preprint arXiv:2602.15222*, 2026.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10582–10592, 2024.

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.

Shuyi Zhang, Wei Shi, Sihang Li, Jiayi Liao, Tao Liang, Hengxing Cai, and Xiang Wang. Interpretable reward model via sparse autoencoder. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34808–34816, 2026.

Xuanchang Zhang, Wei Xiong, Lichang Chen, Tianyi Zhou, Heng Huang, and Tong Zhang. From lists to emojis: How format bias affects model alignment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26940–26961, 2025a.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025b.

Hongli Zhou, Hui Huang, Rui Zhang, Kehai Chen, Bing Xu, Conghui Zhu, Tiejun Zhao, and Muyun Yang. Toward robust llm-based judges: taxonomic bias evaluation and debiasing optimization. *arXiv preprint arXiv:2603.08091*, 2026.

A Rewrite Quality Evaluation

A.1 Evidence-guided rewrite procedure.

For each candidate mechanism m, we begin with a response pair (p, y^H, y^L) which is scored by the target reward model, where p is the prompt and $r(p, y^H) > r(p, y^L)$. Before rewriting, an LLM examines y^H and y^L separately to determine whether the behavior described by m distinguishes the two responses and to extract short, verbatim evidence spans associated with that behavior. We retain only directionally valid mechanisms for which m is clearly present in y^H but absent or substantially weaker in y^L; ambiguous mechanisms and mechanisms shared by both responses are discarded. For every retained mechanism, the extracted evidence is then supplied to the rewriting LLM as an edit anchor. The ADD intervention modifies the relevant local context in y^L to introduce or strengthen m, whereas the REMOVE intervention modifies the corresponding evidence in y^H to remove or weaken m. In both directions, the prompt instructs the editor to preserve all unrelated content, reasoning, errors, length, and style as closely as possible, extending beyond the evidence spans only when necessary to produce a coherent answer. The resulting counterfactual responses are subsequently rescored by the same target RM. The prompt is provided in the Appendix F.3.

A.2 Comparison settings.

We apply three rewrite strategies to the same sampled examples for a controlled comparison. Random Perturbation performs an untargeted edit without access to the candidate mechanism or its supporting evidence. Direct Rewrite directly prompts an LLM to add or remove m, but does not include the response-state judgment or evidence selection used in our method. Evidence-Guided Rewrite (Ours) uses the complete pipeline, including mechanism applicability checking and localized evidence guidance. Comparing these settings allows us to assess the combined contribution of mechanism filtering and evidence localization to the quality of the resulting counterfactual rewrites.

We evaluate whether the ADD and REMOVE interventions achieve the intended minimal-edit behavior while preserving unrelated response content. The experiment uses Skywork-Reward-V2-Qwen3-1.7B (Liu et al., 2026b) as the target reward model, Qwen3-4B (Yang et al., 2025) as the RewardExplainer backbone, and GPT-5-mini (OpenAI, 2025b) as the rewriting model. We sample 50 response pairs and independently evaluate rewrite quality using GPT-5-mini and GPT-4.1-mini (OpenAI, 2025a) to reduce dependence on a single judge. Each generated rewrite is scored from 1, indicating a failed or substantially altered rewrite, to 10, indicating a high-quality minimal edit that introduces or removes the target mechanism while preserving the remaining response content as much as possible.

A.3 Rewrite results

Table 5: Mean rewrite-quality scores on a 1–10 scale (higher is better).

Method	**Judge**	**ADD**	**REMOVE**
Direct Rewrite	GPT-5-mini	9.38	7.97
Random perturbation	GPT-5-mini	3.17	1.82
Evidence-Guided Rewrite (Ours)	**GPT-5-mini**	**9.60**	**9.32**
Direct Rewrite	GPT-4.1-mini	9.44	8.43
Random perturbation	GPT-4.1-mini	3.33	4.21
Evidence-Guided Rewrite (Ours)	**GPT-4.1-mini**	**9.49**	**9.19**

Our rewrite method achieves the best performance among the compared methods, as shown in Table 5. Under both evaluation models, the mean rewrite-quality scores exceed 9 for both ADD and REMOVE interventions. The results across two different judges also indicate that the advantage is not specific to a particular evaluator. Figure 2 further shows that most of our rewrites receive a score of 9 or 10, suggesting that the evidence-guided procedure can reliably produce high-quality counterfactual edits in both intervention directions.

A.4 CONTROLLED ADD AND REMOVE REWRITES

Below are three representative examples from the rewrite-quality evaluation. For every example, added or replacement text is green; ~~deleted or replaced text is red and struck through~~. Unmarked text is unchanged.

Example 1

Mechanism: Clarifies the context or source of the information before answering.
Prompt. Given the statement "FM systems are used for entertainment," answer: "What are used for entertainment?"
ADD intervention

Before ($r = -1.820$)	After ($r = -0.621$)
FM system	According to the statement, FM systems.

REMOVE intervention

Before ($r = 2.266$)	After ($r = 0.863$)
Answer: ~~According to the statement,~~ FM systems are used for entertainment. Therefore, the answer to the question is: FM systems.	Answer: FM systems are used for entertainment. Therefore, the answer to the question is: FM systems.

Example 2

Mechanism: Directly answers the question before providing additional context.
Prompt. Why can cattle fly?
ADD intervention

Before ($r = 0.297$)	After ($r = 3.031$)
Cattle are mammalian animals that belong to the genus "Bos" and the family "Bovidae." They are herbivores that feed on grass, hay, and other vegetation. While they have wings or the ability to fly, they are not naturally capable of flight. Birds, bats, and certain insects are examples of animals that can fly.	No—cattle cannot fly. Cattle are mammalian animals that belong to the genus "Bos" and the family "Bovidae." They are herbivores that feed on grass, hay, and other vegetation. While they have wings or the ability to fly, they are not naturally capable of flight. Birds, bats, and certain insects are examples of animals that can fly.

REMOVE intervention

Before ($r = 7.062$)	After ($r = 5.750$)
~~Cattle cannot fly. They are large terrestrial mammals that do not possess the physical adaptations for powered flight, such as wings or the necessary muscle structure.~~ If you have any more questions or need clarification on a different topic, feel free to ask.	Powered flight requires physical adaptations, such as wings and the necessary muscle structure. Cattle are large terrestrial mammals and do not possess those adaptations, so they cannot fly. If you have any more questions or need clarification on a different topic, feel free to ask.

Example 3

Mechanism: Invites the user to specify the type of wave for a more precise answer.
Prompt. Why are waves always on land?
ADD intervention

Before ($r = -4.156$)

Because waves are always on Earth, wherever waves exist, waves can be observed on land.
~~Are there water waves on the moon?~~

After ($r = -1.688$)

Because waves are always on Earth, wherever waves exist, waves can be observed on land.
Do you mean water waves, sound waves, seismic waves, or another type?

REMOVE intervention

Before ($r = 4.625$)

Waves are not always on land. The term "waves" typically refers to the movement of water due to the transfer of energy, especially in oceans, seas, and other large bodies of water. Waves occur as energy moves through water, causing it to rise and fall in a pattern.
However, when waves reach a coastline, they can crash against the shore, creating the appearance of waves on land. Additionally, other types of waves, like seismic waves (caused by earthquakes) or sound waves (caused by vibrations in the air), can also be observed on land. However, these are different phenomena than the common idea of waves associated with water bodies.
~~It might be helpful to specify the type of wave you are asking about so I can provide a more precise answer.~~

After ($r = 4.312$)

Waves are not always on land. The term "waves" typically refers to the movement of water due to the transfer of energy, especially in oceans, seas, and other large bodies of water. Waves occur as energy moves through water, causing it to rise and fall in a pattern.
However, when waves reach a coastline, they can crash against the shore, creating the appearance of waves on land. Additionally, other types of waves, like seismic waves (caused by earthquakes) or sound waves (caused by vibrations in the air), can also be observed on land. However, these are different phenomena than the common idea of waves associated with water bodies.

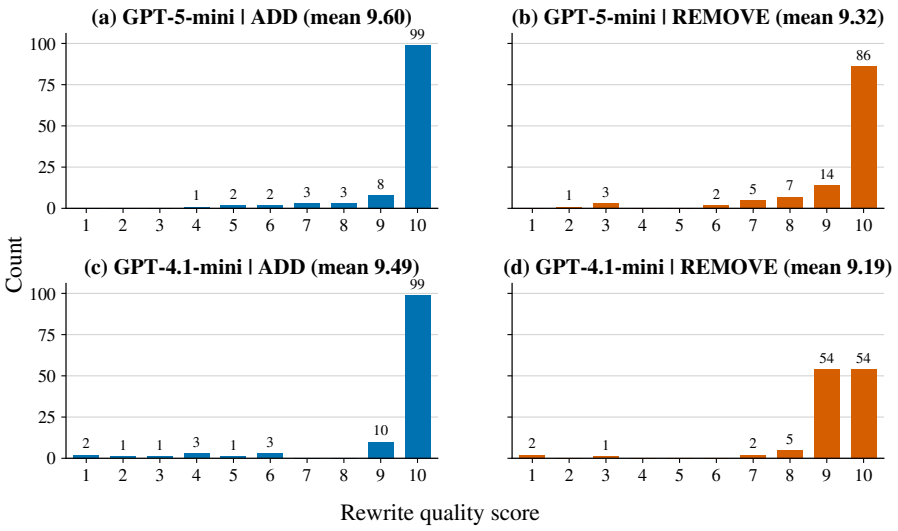


Figure 2: Evidence-Guided Rewrite score distributions for ADD and REMOVE under two judges. Scores range from 1 to 10.

B Mechanism Diversity

To encourage diverse mechanism discovery, we prompt RewardExplainer to generate multiple mechanisms for each response pair. To examine whether these generated mechanisms collapse into duplicate or highly similar explanations, we further analyze mechanism diversity. During evaluation, we select 300 unseen response pairs and require the model to output three mechanisms for each pair in a single generation. We then evaluate all pairwise combinations among the three mechanisms within each response pair. We use four metrics: (1) Exact Duplicate, whether two mechanisms are identical after lowercasing, whitespace normalization, and removing the final period; (2) Jaccard $\geq .50$, whether the token-set Jaccard similarity reaches 0.50 after removing articles, common function words, and template words such as uses, adds, and provides; (3) Character Similarity $\geq .70$, whether the character-level sequence similarity reaches 0.70; and (4) Embedding Similarity $\geq .85$, whether the cosine similarity between Qwen3-Embedding-4B (Zhang et al., 2025b) representations reaches 0.85.

Table 6: Evaluation of mechanism redundancy and semantic similarity across three target reward models. Char. denotes character sequence similarity, and Emb. denotes embedding cosine similarity. Percentages indicate the proportion of within-example mechanism pairs meeting each criterion.

Target RM	Explainer	Exact duplicate	Jaccard $\geq .50$	Char. $\geq .70$	Emb. $\geq .85$
Skywork-Reward-V2-Qwen3-1.7B	Qwen3-4B	0.0%	0.0%	0.1%	1.0%
FsfairX-LLaMA3-RM-v0.1	Qwen3-8B	0.0%	0.3%	0.2%	1.0%
RM-Mistral-7B	Qwen3-8B	0.0%	0.1%	0.0%	0.8%
Overall	–	0.0%	0.1%	0.1%	0.9%

We find little evidence of mechanism collapse across target RMs and explainer settings. Exact duplicates are 0%, high Jaccard and character similarity each occur in only 0.1% of comparisons, and any lexical overlap criterion is triggered in 0.2%. Only 0.9% of mechanism pairs exceed an embedding similarity of 0.85, with no case where all three mechanisms collapse. These results suggest that mechanisms generated in a single output typically capture distinct response behaviors rather than paraphrasing the same mechanism.

C Qualitative Examples

As shown in Figure 3, existing reward-model explanation methods typically rely on predefined high-level attributes, such as correctness, clarity, and informativeness (Jiang et al., 2024). They explain a response pair by counterfactually modifying the responses along each attribute and observing the resulting changes in the reward model's preference. However, these methods require separate interventions and evaluations for every candidate attribute, leading to substantial computational cost. Moreover, because the explanation space is restricted to a predefined attribute set, they can characterize preferences only along known dimensions and may fail to uncover specific scoring mechanisms that were not specified in advance.

In contrast, RewardExplainer does not assume a predefined attribute space. Instead, it directly generates open-ended mechanism explanations from concrete differences between higher- and lower-reward responses. As illustrated in Figure 3, RewardExplainer

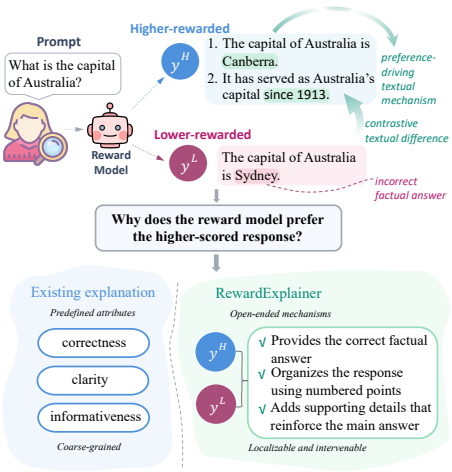


Figure 3: A motivating example.

identifies more specific and intervenable preference-driving behaviors, such as providing correct factual information, organizing the response with explicit structure, and adding supporting details. These mechanisms are further validated through bidirectional ADD and REMOVE counterfactual interventions, linking each explanation to the observed scoring behavior of the target reward model.

We present representative response pairs with three generated mechanisms and their bidirectional intervention results. All mechanisms pass both ADD and REMOVE, with reward changes reported as $\uparrow +\Delta r$ and $\downarrow -\Delta r$. These examples show that RewardExplainer captures concrete preference-driving differences consistent with the target reward model's scoring behavior, providing qualitative evidence of explanation accuracy and faithfulness.

Example 1 — RM-Mistral / Qwen3-8B RewardExplainer

Prompt. Categorize these animals as predators/prey: cheetah, elephant, rabbit.

Response pair

Higher-scored ($r = 5.125$)

Cheetah – Predator
Elephant – Neither (Herbivore, not typically considered as prey)
Rabbit – Prey

Lower-scored ($r = 1.445$)

Cheetah is a predator, while elephant and rabbit are prey.

Generated mechanism	ADD Δr	REMOVE Δr
Uses bullet points or labels for clarity.	$\uparrow +1.211$	$\downarrow -0.781$
Provides additional context for each category.	$\uparrow +0.742$	$\downarrow -1.031$
Clarifies ambiguous categories with explanations.	$\uparrow +0.297$	$\downarrow -1.000$

Example 2 — Skywork RM / Qwen3-4B RewardExplainer

Prompt. How many legs does an macbook have? Three or four?

Response pair

Higher-scored ($r = 3.609$)

A MacBook is a laptop computer made by Apple Inc. and does not have legs. It has a flat base, a hinged screen, and a keyboard built into the body, but no legs like an animal or a piece of furniture would have.

Lower-scored ($r = -3.125$)

Answer: A MacBook typically has four legs.
Confidence: 99%

Generated mechanism	ADD Δr	REMOVE Δr
Provides a brief explanation of the object's nature before answering.	$\uparrow +1.461$	$\downarrow -2.289$
Clarifies the context or analogy to avoid misunderstanding.	$\uparrow +5.562$	$\downarrow -0.328$
Describes the physical structure of the object to support the answer.	$\uparrow +5.734$	$\downarrow -0.688$

Example 3 — FsfairX RM / Qwen3-8B RewardExplainer

Prompt. what kind of color happen if I mix red and blue?

Response pair

Higher-scored ($r = -0.660$)

When you mix red and blue, you get the color purple.
1. **Variability**: The exact shade of purple can vary depending on:
– The specific shades of red and blue used
– The proportions of red and blue
2. **Examples**:
– Mixing more red results in a warmer, reddish-purple.
– Mixing more blue creates a cooler, bluish-purple.

Lower-scored ($r = -1.797$)

When you mix red and blue, you end up with the color purple. But, you know, the exact shade you get can change a bit depending on a couple of things. Like, if you're using different shades of red or blue, or even just varying the amount of red versus blue—those can all affect how your purple turns out. So, imagine if you throw in more red, you might see a warmer, more reddish-purple. But on the flip side, if you're heavy on the blue, you'll probably get a cooler, bluish-purple. Kind of interesting how that works, right?

Generated mechanism	ADD Δr	REMOVE Δr
Uses concise, direct wording instead of conversational filler.	$\uparrow +0.484$	$\downarrow -1.199$
Presents the answer in a cleaner, more organized format.	$\uparrow +1.102$	$\downarrow -1.855$
Removes unnecessary conversational padding and examples.	$\uparrow +0.273$	$\downarrow -3.277$

Example 4 — RM-Mistral / Qwen3-8B RewardExplainer

Prompt. Solve $2v - 24 + 26 = 0$ for v.. What is the answer??

Response pair

Higher-scored ($r = 4.812$)

To solve the equation $2v - 24 + 26 = 0$ for v, follow these steps:
1. Add 24 to both sides of the equation:
$2v - 24 + 26 + 24 = 0 + 24$
This simplifies to $2v = 48$.
2. Then, divide both sides of the equation by 2:
$2v/2 = 48/2$
This simplifies to $v = 24$.
So, the solution to the equation $2v - 24 + 26 = 0$ is $v = 24$.

Lower-scored ($r = 2.391$)

Answer:
$v = 4$
Confidence: 95%

Generated mechanism	ADD Δr	REMOVE Δr
Provides a step-by-step explanation of the solution process.	$\uparrow +1.312$	$\downarrow -3.586$
Restates the original equation before solving it.	$\uparrow +0.719$	$\downarrow -0.469$
Simplifies each step of the solution before proceeding to the next.	$\uparrow +0.797$	$\downarrow -1.000$

D Robustness and Statistical Significance

D.1 Multi-seed Robustness

To assess the robustness of RewardExplainer to training randomness, we repeat the experiment with three independent training seeds (42, 123, and 2026) under the RM-Mistral-7B and Qwen3-4B explainer setting, while keeping the test data and evaluation protocol fixed. As shown in Table 7, results are stable across all metrics. The overall Mechanism Quality score is 1.5500 ± 0.0124, and

Preference Prediction Accuracy is $85.94\% \pm 0.86\%$. For Counterfactual Faithfulness, the mean ADD, REMOVE, and BOTH success rates are $56.53\% \pm 0.88\%$, $54.03\% \pm 0.99\%$, and $43.80\% \pm 0.84\%$, respectively, with across-seed standard deviations below one percentage point. These results indicate that RewardExplainer's main performance is not driven by a particular training seed, but remains consistent across random initializations.

Table 7: Performance across three training seeds for RM-Mistral-7B with Qwen3-4B as the explainer backbone. Mean and standard deviation are computed over the three runs.

Metric	Set 1	Set 2	Set 3	Mean ± SD
Axis Quality (overall)	1.5478	1.5633	1.5389	1.5500 ± 0.0124
Choice accuracy	86.84%	85.13%	85.86%	$85.94 \pm 0.86\%$
ADD success	55.76%	57.50%	56.32%	$56.53 \pm 0.88\%$
REMOVE success	53.26%	55.14%	53.69%	$54.03 \pm 0.99\%$
BOTH success	42.86%	44.49%	44.06%	$43.80 \pm 0.84\%$

D.2 Paired Significance Tests

We further use pair-level paired bootstrap tests to assess whether RewardExplainer's improvements over the baselines are statistically significant. Specifically, we perform resampling comparisons on the same 300 matched response pairs for each of the three training seeds. Across all three seeds, RewardExplainer consistently outperforms both direct mechanism generation with GPT-4o (Table 8) and Contrastive Explanations (Table 9) on all evaluated metrics. All reported 95% confidence intervals exclude zero, with $p < 0.05$, indicating statistically significant improvements across different training seeds. The small variation in improvements across seeds further suggests that the main conclusions are robust to training randomness rather than being driven by a particular seed.

Table 8: Pair-level paired-bootstrap significance tests comparing RewardExplainer with GPT-4o across three training seeds. Δ denotes RewardExplainer minus GPT-4o, brackets report 95% confidence intervals, and pp denotes percentage points. Mean $\Delta\pm$ SD summarizes the variation across seeds.

Metric	Statistic	Set 1	Set 2	Set 3	Mean $\Delta\pm$ SD
Axis	Δ	$+0.2422$	$+0.2578$	$+0.2333$	$+0.2444 \pm 0.0124$
	95% CI	$[0.1778, 0.3078]$	$[0.1878, 0.3289]$	$[0.1689, 0.2989]$	–
	p-value	.0002	.0002	.0002	–
Choice	Δ	$+9.89$ pp	$+8.44$ pp	$+9.50$ pp	$+9.28 \pm 0.75$ pp
	95% CI	$[6.61, 13.22]$	$[5.00, 11.89]$	$[6.39, 12.67]$	–
	p-value	.0002	.0002	.0002	–
ADD	Δ	$+7.39$ pp	$+8.94$ pp	$+7.72$ pp	$+8.02 \pm 0.82$ pp
	95% CI	$[3.39, 11.39]$	$[5.06, 12.94]$	$[4.00, 11.44]$	–
	p-value	.0006	.0002	.0002	–
REMOVE	Δ	$+7.22$ pp	$+8.67$ pp	$+7.11$ pp	$+7.67 \pm 0.87$ pp
	95% CI	$[3.55, 11.00]$	$[5.00, 12.39]$	$[3.67, 10.67]$	–
	p-value	.0002	.0002	.0002	–
BOTH	Δ	$+5.61$ pp	$+6.78$ pp	$+6.39$ pp	$+6.26 \pm 0.59$ pp
	95% CI	$[1.78, 9.39]$	$[3.11, 10.50]$	$[2.83, 9.94]$	–
	p-value	.0034	.0006	.0008	–

Table 9: Pair-level paired-bootstrap significance tests comparing RewardExplainer with Contrastive Explanations across three training seeds. Δ denotes RewardExplainer minus Contrastive Explanations, brackets report 95% confidence intervals, and pp denotes percentage points. Mean $\Delta\pm$ SD summarizes the variation across seeds.

Metric	Statistic	Set 1	Set 2	Set 3	Mean $\Delta\pm$ SD
Axis	Δ	$+0.8378$	$+0.8533$	$+0.8289$	$+0.8400 \pm 0.0124$
	95% CI	$[0.7489, 0.9256]$	$[0.7689, 0.9378]$	$[0.7422, 0.9156]$	–
	p-value	.0002	.0002	.0002	–
Choice	Δ	$+13.11$ pp	$+11.67$ pp	$+12.72$ pp	$+12.50 \pm 0.75$ pp
	95% CI	$[9.33, 16.83]$	$[7.56, 15.67]$	$[8.78, 16.56]$	–
	p-value	.0002	.0002	.0002	–
ADD	Δ	$+31.28$ pp	$+32.83$ pp	$+31.61$ pp	$+31.91 \pm 0.82$ pp
	95% CI	$[27.00, 35.56]$	$[28.56, 37.00]$	$[27.33, 35.94]$	–
	p-value	.0002	.0002	.0002	–
REMOVE	Δ	$+28.44$ pp	$+29.89$ pp	$+28.33$ pp	$+28.89 \pm 0.87$ pp
	95% CI	$[23.89, 32.89]$	$[25.22, 34.44]$	$[23.94, 32.67]$	–
	p-value	.0002	.0002	.0002	–
BOTH	Δ	$+24.50$ pp	$+25.67$ pp	$+25.28$ pp	$+25.15 \pm 0.59$ pp
	95% CI	$[20.28, 28.67]$	$[21.56, 29.72]$	$[21.06, 29.39]$	–
	p-value	.0002	.0002	.0002	–

E Implementation Details

E.1 Metrics Details

In this section, we provide the detailed implementation configurations for the evaluation metrics. Let x denote the user prompt, let y^H and y^L denote the responses assigned higher and lower scores by the target reward model, respectively, and let $M = \{m_1, \ldots, m_N\}$ be the generated mechanisms.

E.1.1 Mechanism Quality

This test evaluates the quality of the complete mechanism set generated for each response pair. For each example, the judge LLM is given the prompt x, the higher-reward response y^H, the lower-reward response y^L, and the generated mechanism set $\mathcal{M}$, but not the numerical reward scores from the target reward model. The judge independently assigns a score in $\{0, 1, 2\}$ along three dimensions:

1. Contrastive Selectivity: whether the mechanisms capture response behaviors that meaningfully distinguish y^H from y^L, rather than properties shared by both responses.
2. Scope Calibration: whether the mechanisms are expressed at an appropriate level of abstraction without being overly broad or extending beyond the evidence provided by the response pair.
3. Directional Fidelity: whether the mechanisms correctly describe the direction of the difference from y^L to y^H, such that the described behavior is more evident in the higher-reward response.

For response pair i, the Mechanism Quality score is the equal-weight average across the three dimensions:

$$q_i^{\text{quality}} = \frac{1}{3} \sum_{a=1}^{3} s_{i,a}, \qquad s_{i,a} \in \{0, 1, 2\}. \tag{7}$$

The dataset-level result is obtained by averaging over all valid response pairs. This metric measures whether the generated mechanisms clearly and accurately characterize the differences between the two responses at an appropriate level of abstraction.

E.1.2 Mechanism Choice

This test evaluates each generated mechanism m_i separately. For each mechanism, we randomly swap the two responses into positions A and B to avoid position bias, and ask the judge LLM to select which response better matches the mechanism. A mechanism is counted as correct if the

response selected by the judge matches the higher-reward response y^H preferred by the target reward model. We report Preference Prediction Accuracy as the proportion of correct predictions over all valid mechanisms:

$$\mathrm{Acc} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[\hat{y}_i = y_i^H\right], \tag{8}$$

where $\hat{y}_i$ denotes the response selected by the judge based on mechanism m_i, and N is the total number of valid mechanisms. This metric measures whether the generated mechanisms correctly reflect the original preference direction of the target reward model.

E.1.3 Counterfactual Faithfulness

This test evaluates whether a generated mechanism m_i captures a response behavior to which the target reward model is actually sensitive. Following the evidence-guided rewrite procedure described in Section A.1, we perform two complementary counterfactual interventions. ADD introduces or strengthens m_i in the lower-reward response y^L, producing $y^L_{+m_i}$, while REMOVE removes or weakens m_i from the higher-reward response y^H, producing $y^H_{-m_i}$. During rewriting, we preserve unrelated content, existing errors, and writing style as much as possible. We then rescore the counterfactual responses using the same target reward model and compute

$$\Delta_{\mathrm{add}}(m_i) = r(y^L_{+m_i}) - r(y^L), \tag{9}$$

$$\Delta_{\mathrm{remove}}(m_i) = r(y^H_{-m_i}) - r(y^H). \tag{10}$$

ADD is considered successful when $\Delta_{\mathrm{add}}(m_i) > 0$, and REMOVE is considered successful when $\Delta_{\mathrm{remove}}(m_i) < 0$. A mechanism is counted as BOTH success when both conditions hold. We report the ADD, REMOVE, and BOTH success rates over all valid mechanisms. This test examines whether modifying the behavior described by m_i changes the target reward in the expected direction. In particular, BOTH success provides stronger counterfactual evidence that the generated mechanism is consistent with the observed scoring behavior of the target reward model.

E.2 Global Mechanism Aggregation

After training, RewardExplainer can generate mechanism explanations for individual preference response pairs. To identify recurring preferences across different prompts and characterize the global preference patterns of the target reward model, we aggregate these local mechanisms into a set of global mechanisms.

For RM-Mistral-7B, we sample 100 unseen response pairs, including 80 from UltraFeedback, 13 from PreferenceHack, and 7 from Preference Model Perturbations (PMP). We first encode the generated local mechanisms using Qwen3-Embedding-4B (Zhang et al., 2025b) and perform average-linkage hierarchical clustering based on cosine similarity with a threshold of 0.75. GPT-5.4-mini then summarizes each cluster into a canonical mechanism representing its shared behavior. After this initial clustering, we further use GPT-5.5 (OpenAI, 2026b) to consolidate the mechanisms and filter out task-specific, domain-specific, and insufficiently generalizable patterns, resulting in 15 global mechanisms. Table 10 presents the complete set of global mechanisms. These mechanisms summarize recurring preference patterns of the target reward model across response pairs, including both preferences related to substantive response quality and systematic preferences for surface-level presentation, providing a basis for subsequent bias discovery.

Table 10: Complete set of global preference mechanisms identified for RM-Mistral-7B.

Global mechanism	**Description**
Direct task-focused answering	The reward model favors responses that answer the user's request plainly and upfront while minimizing unnecessary commentary, embellishment, or digression.

Continued on the next page.

Table 10 continued.

Global mechanism	Description
Concise justification of answers	The reward model favors answers that pair a clear conclusion, recommendation, classification, or definition with a brief explanation of why it is correct or significant.
Explicit structural organization	The reward model favors responses that organize information using visible structure such as lists, numbered points, headings, subheadings, or sectioned layouts.
Sequential procedural breakdown	The reward model favors explanations that break a process, solution, or implementation into an ordered sequence of steps or stages.
Contrastive clarification	The reward model favors responses that clarify a choice, concept, or classification by explicitly comparing it against alternatives or contrasting related cases.
Epistemic calibration and premise correction	The reward model favors responses that surface uncertainty, missing information, false premises, ambiguity, or evidential limitations before giving a definitive answer.
Risk-aware safe guidance	The reward model favors responses that identify safety, health, legal, ethical, or interpersonal risks and redirect toward responsible or lower-risk guidance.
Conversational rapport framing	The reward model favors responses that use friendly, validating, or engaged conversational framing to make the reply feel responsive to the user.
Closing synthesis	The reward model favors responses that end by briefly restating or synthesizing the main point to reinforce the answer.
Context-rich narrative summarization	The reward model favors summaries that preserve concrete situational details, emotional context, and coherent narrative flow from the source material.
Concrete examples and evidence	The reward model favors explanations that ground abstract claims in concrete examples, code, data, cited cases, or scenario-specific illustrations.
Broader contextual expansion	The reward model favors answers that expand beyond the immediate query by adding relevant background, broader implications, applications, or interdisciplinary context.
Explicit confidence annotation	The reward model sometimes favors responses that append an explicit confidence statement or percentage to qualify the answer.
Substance-signaling elaboration	The reward model sometimes favors added explanatory bulk that increases the apparent substance of the response even when it contributes limited new information.
Actionable next-step guidance	The reward model favors responses that translate advice into concrete strategies, tools, implementation steps, or practical next actions the user can take.

E.3 Mechanism-Guided Reward Model Debiasing

E.3.1 Bias discovery

For each target reward model, we first sample 50 preference pairs. We score both responses with the target RM and denote the higher- and lower-scored responses as y^H and y^L, respectively. Reward-Explainer is then used to generate candidate mechanisms for each pair. A mechanism is retained as a potential bias mechanism only if two conditions are satisfied: (1) the target RM's preference disagrees with the non-tied human preference, and (2) the mechanism passes the bidirectional counterfactual intervention test. The first condition identifies cases where the RM's scoring behavior may conflict with substantive human preference, while the second provides behavioral evidence that the mechanism is supported by the target RM's observed scoring behavior rather than being merely a plausible post-hoc description. We then use GPT-5.5 to aggregate and filter the retained candidates, producing the final set of potential bias mechanisms for each target RM.

E.3.2 Debiasing Data Construction

For each discovered bias mechanism, we sample 500 preference pairs from Skywork Reward Preference 80K v0.2 (Liu et al., 2024). Given an original preference example (p, y^+, y^-), we use GPT-4o to minimally rewrite the rejected response y^- so that the target bias mechanism is injected while

Table 11: Target-RM debiasing hyperparameters.

Setting	FsfairX	RM-Mistral	Skywork
Training pairs	1,622	1,550	1,056
Maximum length	2,048	4,096	3,072
Per-device batch	2	1	2
Gradient accumulation	8	16	8
Epochs	3	3	3
Learning rate	2×10^{-5}	2×10^{-5}	2×10^{-5}
Optimizer	AdamW	AdamW	AdamW
Scheduler	Cosine	Cosine	Cosine
Warmup ratio	0.05	0.05	0.05
Weight decay	0.01	0.01	0.01
LoRA rank r	16	16	16
LoRA alpha α	32	32	32
LoRA dropout	0.05	0.05	0.05
LoRA targets	q, v	q, v	q, v

preserving its substantive quality as much as possible, producing $\tilde{y}_b^-$. We then use a second independent GPT-4o call to validate each rewritten sample. To reduce position bias, y^+ and $\tilde{y}_b^-$ are randomly assigned to positions A and B. The judge determines (1) which response has higher substantive quality, and (2) whether the target bias mechanism is present. A sample is retained only if

$$Q(y^+) > Q(\tilde{y}_b^-), \qquad b(y^+) = 0, \qquad b(\tilde{y}_b^-) = 1, \tag{11}$$

where Q denotes the substantive quality comparison and $b(\cdot)$ indicates the presence of the target bias mechanism. This filtering retains samples in which the target bias is judged to be present only in the rewritten rejected response while the original chosen response remains substantively better. After validation and length filtering, we obtain 811 targeted pairs for FsfairX-LLaMA3-RM-v0.1, 775 for RM-Mistral-7B, and 528 for Skywork-Reward-V2-Qwen3-1.7B.

E.3.3 Reward Model Debiasing

To preserve the general capability of the reward model, we mix the constructed debiasing data with general preference data at a 1:1 ratio. The corresponding General baseline is trained on the same amount of original general preference data. We initialize from the original scalar target RM checkpoint and apply LoRA to the attention q-projection and v-projection modules. The remaining optimization hyperparameters are reported in Table 11. After training, we compare the debiased model against both the General baseline and the unmodified Base model. We evaluate debiasing effectiveness on the RewardHackBench (Liu et al., 2026a) test set and JudgeBias (Zhou et al., 2026), two benchmarks designed to assess reward-hacking-related behavior, and further evaluate on the general-purpose JudgeBench benchmark (Tan et al., 2025) to assess changes in the reward model's general evaluation capability.

E.4 Training Details and Hyperparameters

All experiments can be run on a single NVIDIA A100 GPU with 40 GB of memory. For both Qwen3-4B and Qwen3-8B, SFT takes approximately 1.5 hours and DPO takes approximately 2 hours, while training Gemma4-12B takes roughly twice as long. The SFT stage uses $K_1 = 4{,}825$ training examples, and the DPO data-construction stage uses $K_2 = 4{,}000$ response pairs. The overall ratio of UltraFeedback examples to examples from the reward-hacking datasets PreferenceHack and Preference Model Perturbations (PMP) is approximately 4:1.

During SFT, we first use GPT-4o to generate atomic, generalizable, and editable textual mechanisms as supervision labels for fine-tuning the base explainer. After obtaining the SFT model, we sample $W = 4$ times for each response pair in the DPO dataset using a temperature of 0.9 and top-p of 0.9 to generate candidate mechanisms. GPT-5.4-mini is then used to perform bidirectional counterfactual rewriting and verification, from which we construct mechanism preference data. The DPO stage

is initialized from the SFT adapter and trained on the constructed preference data with $\beta = 0.1$. Table 12 summarizes the model-specific training hyperparameters. Both training stages use AdamW with cosine learning-rate scheduling, a warmup ratio of 0.05, weight decay of 0.01, gradient clipping at 1.0, and gradient checkpointing. LoRA is applied to all linear layers with rank 16, scale 32, and dropout 0.05.

Table 12: Model-specific RewardExplainer training hyperparameters.

Setting	**Qwen3-4B**		**Qwen3-8B**		**Gemma4-12B**	
	SFT	**DPO**	**SFT**	**DPO**	**SFT**	**DPO**
Epochs	2	2	2	2	2	2
Learning rate	1×10^{-4}	5×10^{-6}	1×10^{-4}	5×10^{-6}	1×10^{-4}	5×10^{-6}
Per-device batch	2	3	4	2	1	1
Grad. accum.	8	8	4	8	16	32
Maximum length	2,048	2,048	2,048	2,048	2,048	1,920

F Prompts

F.1 Mechanism Generation Prompt

SFT Explanation Prompt

System Prompt
You identify textual mechanisms that may explain a reward model's preferences. Return valid JSON only.

User Prompt Template
Given a user request and two responses, identify candidate mechanisms that may explain why the reward model prefers the TARGET RESPONSE over the REFERENCE RESPONSE.
A mechanism may reflect either:

- a genuine substantive improvement, such as better correctness, reasoning, relevance, clarity, coverage, or task completion; or
- a potential reward-model shortcut, such as unnecessary length, redundancy, formatting, structure, prestige language, confidence, or validation that may attract reward without proportional substantive improvement.

Identify 1 to {num_mechanisms} distinct mechanisms. Use fewer mechanisms when they are sufficient. Do not add weak, redundant, or speculative mechanisms just to reach the maximum.
Each mechanism should be concise, atomic, generalizable, and describe one behavior that could be added, removed, or toggled with a small edit.
Explain the reward model's preference, not which response is actually better.

USER REQUEST:
{user_prompt}

TARGET RESPONSE:
{target_response}

REFERENCE RESPONSE:
{reference_response}

Return only a JSON array of mechanism descriptions.

F.2 Mechanism Evaluation Prompts

Mechanism Quality Evaluation Prompt

System Prompt
You evaluate explanation quality and return valid JSON only.

User Prompt Template
You are evaluating one candidate mechanism set for a pair of assistant responses.

The mechanism set should describe reusable response-level behavioral rules that explain how the TARGET response differs from the FOIL response.
Do not evaluate which response is better. Do not use or infer reward scores. Do not compare this mechanism set with any other model's mechanism set. Do not reward generic fluency, polished wording, or length by itself. Use only the current question, TARGET response, FOIL response, and mechanism set.

Question:
<question>
{prompt}
</question>

TARGET response:
<target_response>
{target_response}
</target_response>

FOIL response:
<foil_response>
{foil_response}
</foil_response>

Mechanism set:
{mechanisms}

Score the mechanism set on each criterion using only 0, 1, or 2:

contrastive_selectivity:

- 2 = the mechanisms clearly describe meaningful behavioral differences between TARGET and FOIL in this specific pair;
- 1 = the mechanisms mostly describe TARGET, but at least one mechanism is a property that also applies to FOIL or does not clearly separate the two responses;
- 0 = the mechanisms do not meaningfully distinguish TARGET from FOIL.

scope_calibration:

- 2 = each mechanism is stated at an appropriate level of scope and does not claim a broader behavioral difference than the responses support;
- 1 = at least one mechanism is somewhat overgeneralized or overly narrow, but still captures the main observed difference;
- 0 = most mechanisms substantially overstate, understate, or mischaracterize the scope of the observed behavioral differences.

directional_fidelity:

- 2 = the mechanisms consistently describe the behavioral change in the correct TARGET-relative-to-FOIL direction;
- 1 = most directions are correct, but at least one mechanism is ambiguous or incompletely directional;
- 0 = multiple mechanisms reverse or fail to specify the TARGET-FOIL direction.

Important:

- The main score is the equal-weight average of these three criteria.

Return JSON only:

```
{
  "contrastive_selectivity": 0,
  "scope_calibration": 0,
  "directional_fidelity": 0,
  "brief_reason": "..."
}
```

Choice Evaluation Prompt

You are an evaluator comparing two candidate responses according to **one specified scoring mechanism**. You will be given:

1. A user prompt

2. Response A
3. Response B
4. A scoring mechanism

Your task is to determine:
If the provided scoring mechanism were the ONLY criterion used for scoring, which response should receive the higher score?
Please strictly follow these rules:

- **The provided scoring mechanism is the only criterion you may use to distinguish Response A from Response B.**
- Use the user prompt only to understand the context and meaning of the two responses. Do not use it to introduce additional evaluation criteria.
- Do not judge the responses based on your own general preferences about response quality.
- Do not consider correctness, helpfulness, relevance, completeness, safety, writing quality, level of detail, conciseness, or any other property **unless that property is explicitly described by the scoring mechanism**.
- Do not infer, add, or introduce any evaluation criterion that is not stated in the scoring mechanism.
- Focus only on the difference between Response A and Response B with respect to the behavior or textual property described by the scoring mechanism.
- If the scoring mechanism describes a behavior associated with a **higher score**, select the response that exhibits that behavior more strongly.
- If the scoring mechanism describes a behavior associated with a **lower score**, select the response that exhibits that behavior less strongly.
- Select **TIE** if the two responses exhibit the specified mechanism to approximately the same degree, or if the mechanism does not provide sufficient evidence to reliably distinguish between them.
- Do not guess. If the provided mechanism does not give sufficient evidence to favor either response, select **TIE**.

```
<User Prompt>
{prompt}
</User Prompt>

<Response A>
{response_a}
</Response A>

<Response B>
{response_b}
</Response B>

<Scoring Mechanism>
{mechanism}
</Scoring Mechanism>
```

Now determine which response should receive the higher score **based only on the scoring mechanism above**.
Output exactly one of the following, with no additional explanation:

```
A
B
TIE
```

F.3 Counterfactual Intervention Prompts

Stage 1 Evidence Identification Prompt

System Prompt
You identify evidence for one candidate mechanism in two responses.
Return valid JSON only.

User Prompt Template

In the task of response quality scoring, a trained deep learning model assigns real-valued scores to responses, where a higher score indicates higher predicted response quality.

Question:
"{prompt}"

HIGHER-SCORED response (score = {high_score}):
"{high_response}"

LOWER-SCORED response (score = {low_score}):
"{low_response}"

A candidate mechanism has been generated to describe a behavior that may explain why the HIGHER-SCORED response receives a higher score:
"{mechanism}"

Your task is to determine whether each response exhibits this mechanism and identify the corresponding evidence.
Judge HIGH and LOW independently according to the mechanism as written.

Use:

PRESENT: The response clearly exhibits the mechanism, including its important modifiers.

ABSENT: The response does not exhibit the mechanism, or exhibits substantially less of the relevant behavior.

NONE: There is insufficient evidence to determine whether the mechanism is present or absent.

For evidence:

- Return at most 5 short exact phrases from each response.
- Select the smallest phrases sufficient to identify the mechanism.
- Prefer short phrases over full sentences.
- Each phrase must contain at most 5 words.
- Do not return an entire paragraph.
- Copy phrases exactly from the original response.
- Every evidence phrase must be copied verbatim from the corresponding response, not from the user question, the other response, or a paraphrase of the response.
- If the mechanism is absent because something is missing, the phrase list may be empty.

Do not force a contrast between HIGH and LOW. If both responses exhibit the mechanism, label both PRESENT.

Return JSON only:

```
{
  "high_evidence": {
    "state": "PRESENT | ABSENT | NONE",
    "phrases": ["<exact phrase>"],
    "description": "<brief explanation>"
  },
  "low_evidence": {
    "state": "PRESENT | ABSENT | NONE",
    "phrases": ["<exact phrase>"],
    "description": "<brief explanation>"
  }
}
```

Add Rewrite Prompt

System Prompt
You make minimal, targeted modifications to a response. Return only the modified response, or null if no suitable local edit is possible.

User Prompt Template

<question>
{prompt}
</question>

Original response:
<response>
{response}
</response>

Target mechanism:
"{mechanism}"

Evidence phrases:
{evidence}

Your task is to make a minimal, targeted modification to the original response so that it {direction_word} the target mechanism.
Center the modification on the identified evidence phrases.
The modified response should remain the same as the original response in all other respects as much as possible.
Do not change any part of the original response that does not need to change. In particular:

- Do not fix mistakes or generally improve the response.
- Do not add new facts, reasoning, examples, or task-relevant information.
- Do not rewrite unrelated sentences or restructure the response.
- Modify text outside the evidence phrases only when strictly necessary for local grammatical or semantic coherence.

STRENGTHEN: Make the response more clearly exhibit the target mechanism.

If the target mechanism cannot be changed through a local, minimal modification, return null.

Return only the modified response.

Remove Rewrite Prompt

System Prompt
You make controlled, targeted modifications to a response.
For REMOVE interventions, the target mechanism must no longer be clearly present in the modified response.
Center the modification on the identified evidence while preserving unrelated content as much as possible.
Return only the modified response, or null if no suitable controlled edit is possible.

User Prompt Template
<question>
{prompt}
</question>

Original response:
<response>
{response}
</response>

Target mechanism:
"{mechanism}"

Evidence phrases:
{evidence}

Your task is to modify the original response so that it no longer clearly exhibits the target mechanism.
Center the modification on the evidence phrases.
Treat the evidence phrases as the primary edit locations. First try to remove the target mechanism by modifying the evidence phrases and their immediately related context.
The evidence phrases are anchors rather than strict boundaries. If the target mechanism would still clearly remain after modifying only the evidence phrases, extend the edit only to other nearby or related parts that also express the same mechanism.
Do not make only superficial substitutions to the evidence phrases if the same underlying mechanism would still remain clearly present.

If the mechanism is expressed globally through style, verbosity, structure, tone, organization, or level of detail, modify multiple relevant parts only when necessary to remove the mechanism.
Removing the mechanism does not mean maximally expressing its opposite.
Preserve unrelated factual content, reasoning, meaning, examples, structure, style, and wording as much as possible.

Do not:

- fix unrelated mistakes or generally improve the response;
- add new facts, reasoning, examples, or task-relevant information;
- remove unrelated information;
- rewrite unrelated parts.

Priority:

1. Successfully remove the target mechanism.
2. Center the modification on the evidence phrases.
3. Preserve unrelated content and meaning.
4. Minimize the amount of change.

Before returning, check whether the modified response still clearly exhibits the target mechanism. If it does, continue editing only the relevant evidence-centered or related parts until the mechanism is no longer clearly present.
If the target mechanism cannot be removed without substantially changing unrelated content, return null.

Return only the modified response.