

ReflectRM: Boosting Generative Reward Models via Self-Reflection within a Unified Judgment Framework

Kai Qin^{1,2,†}, Liangxin Liu², Yu Liang², Longzheng Wang², Yan Wang², Yueyang Zhang², Long Xia², Zhiyuan Sun², Houde Liu^{1,‡}, Daiting Shi^{2,‡},

¹Shenzhen International Graduate School, Tsinghua University

²Baidu Inc., Beijing, China

tank24@mails.tsinghua.edu.cn, {liuliangxin, liangyu05}@baidu.com

{wanglongzheng, wangyan78, zhangyueyang, xialong01}@baidu.com

{sunzhiyuan01, shidaiting01}@baidu.com, liu.hd@sz.tsinghua.edu.cn

Abstract

Reward Models (RMs) are critical components in the Reinforcement Learning from Human Feedback (RLHF) pipeline, directly determining the alignment quality of Large Language Models (LLMs). Recently, Generative Reward Models (GRMs) have emerged as a superior paradigm, offering higher interpretability and stronger generalization than traditional scalar RMs. However, existing methods for GRMs focus primarily on outcome-level supervision, neglecting analytical process quality, which constrains their potential. To address this, we propose **ReflectRM**, a novel GRM that leverages self-reflection to assess analytical quality and enhance preference modeling. ReflectRM is trained under a unified generative framework for joint modeling of response preference and analysis preference. During inference, we use its self-reflection capability to identify the most reliable analysis, from which the final preference prediction is derived. Experiments across four benchmarks show that ReflectRM consistently improves performance, achieving an average accuracy gain of +3.7 on Qwen3-4B. Further experiments confirm that response preference and analysis preference are mutually reinforcing. Notably, ReflectRM substantially mitigates positional bias, yielding +10.2 improvement compared with leading GRMs and establishing itself as a more stable evaluator. Our code is available at <https://github.com/yuliangCarmelo/ReflectRM>.

1 Introduction

The remarkable advancements in LLMs have endowed them with exceptional multi-tasking capabilities, transforming how humans approach complex problem-solving and decision-making (Achiam et al., 2023; Jaech et al., 2024; Liu et al., 2024a; Team et al., 2025; Guo et al., 2025a; El-Kishky

[†]Work done while interning at Baidu.

[‡]Corresponding author.

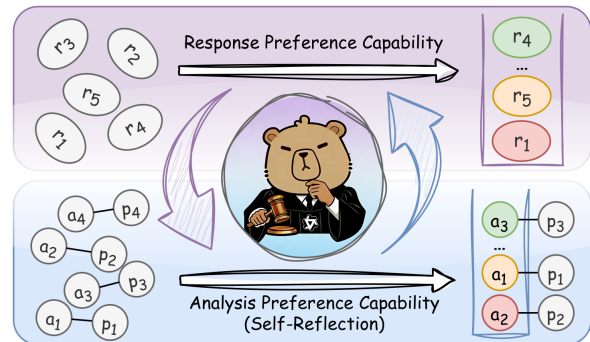


Figure 1: Mutual reinforcement in ReflectRM. By unifying response preference and analysis preference into a single conditional generative process, ReflectRM internalizes a more robust and reliable evaluative logic.

et al., 2025). To better align these models with human values, RLHF has become a key paradigm, as Supervised Fine-Tuning (SFT) alone often fails to capture the nuanced spectrum of human preferences (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022; Dong et al., 2024; Lambert, 2025; Li et al., 2025; Seed et al., 2025). Within the RLHF pipeline, RMs play a fundamental role by providing the training signal that ultimately determines the alignment quality of LLMs. Recently, the research focus has shifted to GRMs, a new paradigm that outputs both textual analysis and preference labels (Mahan et al., 2024; Gunjal et al., 2025; Dineen et al., 2025; Wang et al., 2025a; Ma et al., 2025; Liu et al., 2025b; Yu et al., 2025; Wu et al., 2025; Ye et al., 2025). Compared to traditional scalar RMs, GRMs demonstrate superior evaluative capabilities and better generalization. These advantages make GRMs pivotal for advancing LLMs toward more general proficiency in handling complex and open-ended tasks.

Despite rapid advancements in GRMs (Zhang et al., 2024; Liu et al., 2025c; Chen et al., 2025a; Whitehouse et al., 2025; Guo et al., 2025b; Chen et al., 2025b), most existing methods focus pri-

marily on outcome supervision, providing little to no direct supervision over the analytical reasoning process. Representative approaches include J1 (Whitehouse et al., 2025), which employs online reinforcement learning to generate reasoning for preference judgments, and RM-R1 (Chen et al., 2025b), which combines reasoning distillation via chain-of-rubrics with reward-based optimization. However, both optimize models based primarily on final verdict correctness, without explicit supervision of the analysis process. Although recent work Deepseek-GRM (Liu et al., 2025c) has explored trajectory supervision, it formulates process evaluation as an external classification task applied only during inference, requiring auxiliary models and a complex multi-stage pipeline. Consequently, how to provide effective analysis process supervision for GRMs remains an open research question.

In this paper, we propose ReflectRM, a framework that leverages self-reflection to assess analytical quality and improve pairwise preference judgment. During training, we create process-level preference pairs (Reflection Data) from standard preference tasks by comparing analytical processes leading to correct versus incorrect outcomes. By integrating standard preference data and reflection data into a unified objective, the model jointly develops two mutually reinforcing capabilities: response preference judgment and analysis preference judgment (self-reflection). During inference, we adopt a two-stage strategy: the model first generates multiple candidate outputs and selects one with high confidence analysis as an anchor. It then employs self-reflection to compare the remaining candidates with this anchor, ultimately selecting the most reliable ones to form the final prediction.

Experimental results across four benchmarks and three model scales demonstrate the consistent effectiveness of ReflectRM. Further analysis reveals a mutually reinforcing relationship between response preference and self-reflection capabilities, which significantly improves the reliability of the model’s core judgments. Moreover, our method effectively mitigates positional bias, leading to substantial gains in positional consistency. In summary, our contributions are as follows:

- We propose a unified judgment framework that models both response preference and analysis preference as a single generative task, enabling the unified training of both capabilities.
- ReflectRM inherently leverages self-reflection

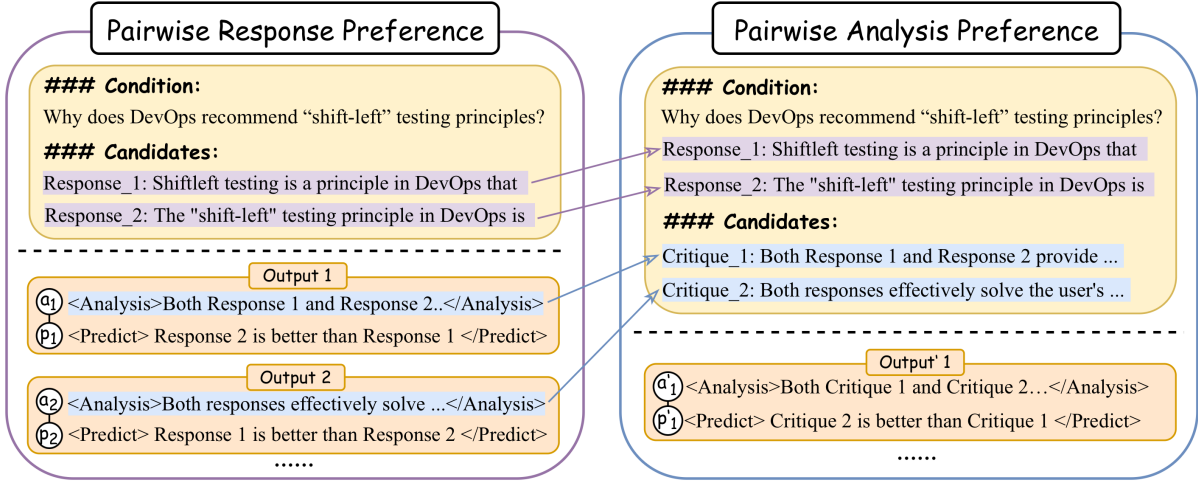
to enhance judgments, without requiring auxiliary models or complex training pipelines.

- Experimental results across four benchmarks and three different models demonstrate the effectiveness of ReflectRM.
- Further analysis reveals that response preference and self-reflection are mutually reinforcing capabilities, leading to substantial mitigation of positional bias.

2 Related Work

Generative Reward Models Recent studies indicate a shift in research focus from traditional scalar RMs toward GRMs. GRMs are regarded as a promising direction in reward modeling, owing to their superior interpretability and generalization capabilities. For instance, DeepSeek-GRM (Liu et al., 2025c) introduces a pointwise generative reward model trained via reinforcement learning to generate detailed critiques and self-derived evaluation rubrics, thereby enabling more flexible and task-agnostic scoring than scalar RMs. Following this generative perspective, RM-R1 (Chen et al., 2025b) proposes a two-stage pipeline that first distills high-quality reasoning traces and then applies reinforcement learning with verifiable reward signals. Similarly, the Reward Reasoning Model (Guo et al., 2025b) extends GRMs by incorporating explicit reasoning steps before final judgment, even without annotated reasoning traces. Despite these rapid advances, current GRM methods remain limited: they primarily provide general rewards based on outcome-level supervision, overlooking the optimization of the analytical process, which in turn constrains further performance improvements.

Self Reflection in LLMs Self-reflection (Renze and Guven, 2024; Liu et al., 2024c) refers to the ability of LLMs to evaluate, critique, and iteratively improve their own outputs. This capability allows models to analyze initial responses in a structured manner, refine them, and ultimately generate answers of higher quality and accuracy. Demonstrating its effectiveness, the TASTE framework (Wang et al., 2024c) achieves significant performance gains in translation by first assessing the quality of its initial output and then refining it based on this assessment. Similarly, SelectIT (Liu et al., 2024d) utilizes self-reflection for high-quality data selection without relying on external models. A



Training under the Unified Framework

Two-Stage Inference via Self-Reflection

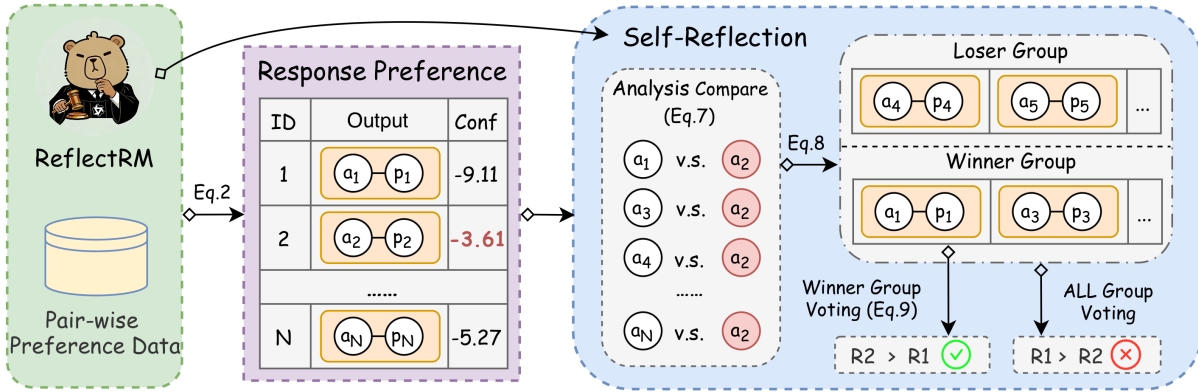


Figure 2: Overview of the ReflectRM method. **(Top)** Our *Unified Judgment Framework*, which models response preference and analysis preference as a single conditional generative process. **(Bottom)** The *Two-Stage Inference Strategy* that leverages the model’s self-reflection capability to identify and aggregate reliable analytical traces, yielding more robust and reliable final judgments.

further advancement is Progressive Self-Reflection (PSR) (Phan et al., 2025), a novel inference-time technique that empowers LLMs to dynamically self-monitor and correct their outputs. While these methods have proven successful for LLMs, their application to GRMs remains unexplored.

Uniqueness of ReflectRM ReflectRM distinguishes itself by unifying response and analytical preferences into a single generative process. Unlike prior GRMs that treat reasoning as an unconstrained byproduct of outcome optimization, ReflectRM explicitly supervises the analytical process within a preference-based architecture. This unified framework ensures that both capabilities stem from a shared evaluative logic, making them mutually reinforcing rather than competing. Furthermore, by eliminating complex multi-stage training pipelines and auxiliary classifiers, ReflectRM offers a stream-

lined and scalable method for advancing generative reward modeling.

3 ReflectRM

Most current reward models are supervised solely based on the correctness of final outcomes while neglecting the explicit supervision of the analytical process. Self-reflection refers to the model’s capability to evaluate the analysis and refine its own reasoning, which has proven effective in enhancing output quality. However, this capability has rarely been integrated into the training of reward models. To bridge this gap, we propose **ReflectRM**, a unified framework that incorporates both process and outcome supervision into reward modeling.

3.1 A Unified Judgment Framework

In this section, we propose a novel framework that unifies response preference and analysis preference

into a single, cohesive generative structure.

Our core idea is to formulate the supervision of the analytical process as a preference task, and frame all preference tasks as a *conditional generative process*. Given a specific condition ϕ and candidates δ , this process aims to generate a comprehensive judgment text, comprising a textual analysis a and a prediction p . This process can be formally expressed as:

$$\underbrace{(a, p)}_O \sim f_\theta(\cdot \mid \underbrace{\phi; \delta}_I) \quad (1)$$

where f_θ represents the LLM parameterized by θ , while $I = (\phi; \delta)$ and $O = (a, p)$ denote the input and output of the model. By unifying different preference tasks under this unified generative formulation, ReflectRM acquires a more fundamental and generalized preference judgment capability. This formulation naturally induces two distinct but mutually reinforcing capabilities simply by varying the conditions ϕ and candidates δ .

Capability 1: Pairwise Response Preference

The first capability derived from our unified framework is the preference of responses, which serves as the core reward modeling capability. This is instantiated from Equation (1) by setting the condition ϕ to the user’s query q and the candidates δ to the two responses (r_1, r_2) :

$$(a, p) \sim f_\theta(\cdot \mid q; r_1, r_2) \quad (2)$$

As illustrated in the "Pairwise Response Preference" box in Figure 2, this directs the model to perform the primary task of judging which response is better while providing a supporting analysis.

Capability 2: Pairwise Analysis Preference (Self-Reflection) Our unified framework also introduces pairwise analysis preference, or **Self-Reflection**, which enables the model to evaluate the quality of its own analytical processes. To achieve this, the condition ϕ in Equation (1) is expanded to the triplet (q, r_1, r_2) to provide the full context, while the candidates δ now become two distinct analytical processes (a_1, a_2) generated via the response-preference task (Equation (2)):

$$(a', p') \sim f_\theta(\cdot \mid q, r_1, r_2; a_1, a_2) \quad (3)$$

This allows the model to act as a meta-judge to evaluate its own analysis. To avoid potential confusion between the model’s input and output, we

refer to the candidate analyses under evaluation as *critiques*, as illustrated in the "Pairwise Analysis Preference" box in Figure 2.

3.2 Training under the Unified Framework

Data Collection To develop the model’s core reward modeling capability, we utilize standard *Preference Data* (abbreviated as **Pref.**). Each instance consists of a user query q , a pair of candidate responses (r_1, r_2) , and a ground-truth label y indicating the preferred response. Furthermore, we propose a novel data format termed *Reflection Data* (abbreviated as **Refl.**) to supervise the quality of the analytical process. Similar to the Chain-of-Thought (CoT) process in reasoning models, the analytical trace in GRMs directly determines the reliability of its judgment. Building on this, we generate multiple outputs for preference data and pair their analyses based on the correctness of their predictions to construct Refl. data.

Unified Training Based on the framework in Section 3.1, we mix the Pref. data and Refl. data for unified training. Our unified framework ensures that these two types of data signals guide the same core preference judgment capability, rather than serving as unrelated objectives. This inherent connection allows us to combine both datasets into a single training process without causing task conflict, enabling ReflectRM to internalize a more robust and reliable evaluative logic.

It is important to note that the supervision signal for Refl. data does not require the model to imitate a perfect or "absolute" analytical process. Instead, it guides the model to identify which analysis is relatively more reasonable when comparing a correct trajectory with an incorrect one, mitigating the risk of rewarding post-hoc rationalizations.

3.3 Two-Stage Inference via Self-Reflection

Following Section 3.2, ReflectRM concurrently acquires response preference and self-reflection capabilities. We leverage these dual capabilities to perform a two-stage inference strategy, as illustrated in the bottom of Figure 2.

Stage 1: Rollouts and Confidence-guided Anchor Selection

Given a query q and response pair (r_1, r_2) , we perform N independent rollouts (where $N = 8$ in our experiments) using the response preference capability defined in Equation (2). This produces an output set $\mathcal{O} = \{O_1, O_2, \dots, O_N\}$, where

each output O_i is a tuple consisting of an analysis and a prediction:

$$O_i = (a_i, p_i), \quad p_i \in \{1, 2\} \quad (4)$$

Specifically, the model’s textual judgment is mapped to the index of the preferred response $\{1, 2\}$, where $p_i = 1$ indicates that r_1 is better than r_2 , and $p_i = 2$ indicates the opposite.

To effectively leverage the self-reflection capability, we select a high-confidence output from \mathcal{O} to serve as a reliable anchor based on the generation probability (Fu et al., 2025). For each output O_i , we identify the bottom 10% of tokens with the lowest log-probabilities within the generated sequence, denoted as T_{bottom} , and calculate the confidence score as follows:

$$\text{Conf}(O_i) = \frac{1}{|T_{\text{bottom}}|} \sum_{t_j \in T_{\text{bottom}}} \log P(t_j) \quad (5)$$

The anchor output O_{anchor} is then defined as the one with the highest confidence score:

$$O_{\text{anchor}} = \underset{O_i \in \mathcal{O}}{\text{argmax}} \text{Conf}(O_i) \quad (6)$$

This method identifies the most internally coherent output to serve as a high-quality baseline from a response-level preference perspective.

Stage 2: Self-Reflection and Voting. Then, we use the self-reflection capability to evaluate whether other candidate analyses are better than the anchor, as formulated in Equation (3). For each output O_i ($i \neq \text{anchor}$), we generate a reflection result (a'_i, p'_i) by treating a_i and a_{anchor} as candidate analyses in a random order:

$$(a'_i, p'_i) \sim f_{\theta}(\cdot \mid q, r_1, r_2; a_i, a_{\text{anchor}}) \quad (7)$$

the result p'_i reveals whether the model prefers analysis a_i over a_{anchor} . We then collect all outputs that outperform the anchor into a *Winner Group* \mathcal{W} :

$$\mathcal{W} = \{O_i \mid i \neq \text{anchor and } O_i \succ O_{\text{anchor}}\} \quad (8)$$

where $O_i \succ O_{\text{anchor}}$ indicates that the reflection result p'_i chooses a_i as the better analysis.

The final prediction P_{final} is determined by a majority vote (Mahan et al., 2024; Wang et al., 2024b) among the outputs in \mathcal{W} , expressed as:

$$P_{\text{final}} = \underset{k \in \{1, 2\}}{\text{argmax}} \sum_{O_i \in \mathcal{W}} \mathbb{I}(p_i = k) \quad (9)$$

where \mathbb{I} is the indicator function. Crucially, if \mathcal{W} is empty or Equation (9) results in a tie, we include the anchor output by setting $\mathcal{W} \leftarrow \mathcal{W} \cup \{O_{\text{anchor}}\}$, and repeat the majority voting process.

The effectiveness of this mechanism is further illustrated in Appendix A.6, where a qualitative case study demonstrates how self-reflection identifies and corrects analytical oversights during inference.

4 Experiments

4.1 Setup

Training Dataset We construct our training data using the HelpSteer3-Preference dataset (Wang et al., 2025b), a large-scale collection of open-ended tasks spanning diverse domains such as STEM, coding, and multilingual scenarios. To build the training set, we first exclude easy cases where the base model consistently selects the correct preferred response across all trials. From the remaining pool, we specifically leverage instances with inconsistent preferences to derive the reflection data (Refl.). The final training set comprises a 4:1 mixture of Pref. data and Refl. data. More details on data statistics and the construction pipeline are provided in Appendix A.1. To further evaluate the generalizability of ReflectRM across different data distributions, we also apply our framework to the Skywork-Reward-V2 dataset (Liu et al., 2025a). Detailed results regarding its performance and positional consistency are provided in Appendix A.3.

Benchmarks To ensure a robust evaluation, we assess **ReflectRM** on five established benchmarks: (1) **JudgeBench** (Tan et al., 2024), which evaluates objective correctness in challenging real-world tasks such as coding and reasoning; (2) **Reward-Bench** (Lambert et al., 2025), a standard for measuring alignment with general human preferences across chat, safety, and reasoning; (3) **RM-Bench** (Liu et al., 2024e), designed to test the model’s ability to distinguish core substance from stylistic distractions; (4) **RMB** (Zhou et al., 2024), a comprehensive suite covering 49 real-world task categories; and (5) **PPE-Preference** (Frick et al., 2024), comprising 16k human-labeled pairs sourced from unfiltered user interactions.

Baselines We compare ReflectRM against three categories of baselines: (1) **Base Model**, the original instructed LLMs without preference fine-tuning; (2) **RFT**, our primary baseline trained exclusively on the preference dataset using rein-

ID	System	Source		RewardBench	RM-Bench	RMB	PPE-Preference	Overall	
		Pref.	Refl.					AVG	Δ (\uparrow)
<i>Open Generative Reward Models</i>									
1	Llama-3-OffsetBias-RM-8B	-	-	89.0	71.3	57.8	59.2	69.3	-
2	ArmoRM-Llama3-8B-v0.1	-	-	90.4	69.3	64.6	60.6	71.2	-
3	Skywork-Reward-Llama-3.1-8B-v0.2	-	-	93.1	72.1	66.6	62.2	73.5	-
4	GRAM-R ² -3B	-	-	-	84.6	-	-	-	-
5	DeepSeek-GRM-27B (MetaRM)	-	-	90.4	-	70.3	67.2	-	-
6	RM-R1-DeepSeek-Distilled-Qwen-32B	-	-	90.9	83.9	69.8	-	-	-
<i>Implemented Existing Method</i>									
7	Qwen3-4B	-	-	84.8	75.4	70.4	60.1	72.7	-
8	4 + RFT	✓	✗	84.0	76.5	71.2	62.5	73.5	0.8
<i>Our Method</i>									
9	4 + Unified Training	✓	✓	85.3	76.5	75.3	63.9	75.2	2.5
10	6 + Two-Stage Inference (ReflectRM)	✓	✓	86.7	77.7	77.1	64.2	76.4	3.7
<i>Implemented Existing Method</i>									
11	Qwen3-8B	-	-	85.0	78.5	73.9	62.9	75.1	-
12	8 + RFT	✓	✗	86.4	77.4	76.0	64.5	76.1	1.0
<i>Our Method</i>									
13	8 + Unified Training	✓	✓	87.9	81.2	73.0	64.3	76.6	1.5
14	10 + Two-Stage Inference (ReflectRM)	✓	✓	89.2	82.7	73.3	64.0	77.3	2.2

Table 1: Main results on four RM benchmarks. Qwen3-4B and Qwen3-8B are chosen as the backbone model. The best results under each backbone are labeled using **bold font**.

forcement fine-tuning; and (3) **Open Generative Reward Models**, which include several leading performance open-source models. Specifically, this category comprises: Llama-3-OffsetBias-RM-8B (Park et al., 2024), ArmoRM-Llama3-8B-v0.1 (Wang et al., 2024a), Skywork-Reward-Llama-3.1-8B-v0.2 (Liu et al., 2024b), GRAM-R²-3B (Wang et al., 2026), DeepSeek-GRM-27B (MetaRM) (Liu et al., 2025c), and RM-R1-DeepSeek-Distilled-Qwen-32B (Chen et al., 2025b). Together, these baselines offer a broad context for evaluating the performance and robustness of our method.

Implementation Details We fine-tune Qwen3-4B and Qwen3-8B (Yang et al., 2025) backbones using the Group Relative Policy Optimization (GRPO) algorithm (Shao et al., 2024), which eliminates the need for a separate value model by estimating advantages within sampled groups. Training is conducted for 3 epochs with a batch size of 64, a learning rate of 1e-6, and a maximum generation length of 1024 tokens. For decoding, we used a sampling temperature of 1.0 to generate $N = 8$ outputs for rollouts, and greedy decoding otherwise for deterministic evaluation. All experiments were conducted on 16 NVIDIA H800 GPUs.

4.2 Main Results

The main performance results of ReflectRM are detailed in Table 1. The results show that while

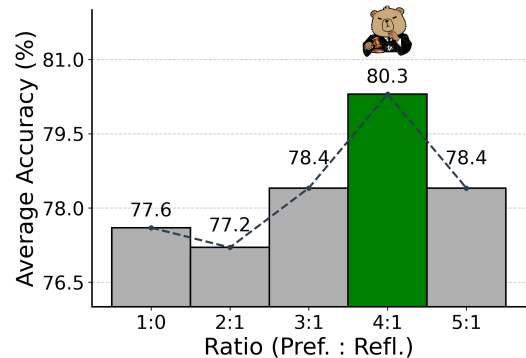


Figure 3: Effect of preference-to-reflection ratio. A 4:1 ratio provides the optimal balance for training.

the standard RFT baseline achieves improvements over the base model, its performance is limited by the absence of process-level supervision (Systems 8 and 12). In contrast, incorporating reflection data into training yields significant performance gains across all model scales (Systems 9 and 13). These results confirm our hypothesis that, within our unified framework, learning to reflect on the analysis process and learning response preferences are mutually reinforcing. The reflection process provides richer supervisory signals, which in turn facilitate more effective learning of preference modeling.

As shown in Table 1, **ReflectRM** achieves the highest performance gains across different model scales, improving average accuracy by 3.7 points

System	R.B.	RMB	AVG	Δ
ReflectRM	86.7	77.1	81.9	-
w/ Random Anchor	86.0	76.8	81.4	-0.5
w/ Random Winners	85.7	76.0	80.9	-1.0

Table 2: Performance comparison on inference strategies. Both a reliable anchor and self-reflection capability are essential for ReflectRM.

System	Pref.	Refl.	Sum	R.B.	RMB	AVG
RFT	13.7k	-	13.7k	84.0	71.2	77.6
w/ Scaled Pref.	17.1k	-	17.1k	85.0	73.2	79.1
w/ Refl. (Ours)	13.7k	3.4k	17.1k	85.3	75.3	80.3

Table 3: Impact of reflection data on response preference. Reflection data offers richer learning signals compared to simply scaling preference data volume.

and 2.2 points for the 4B and 8B models, respectively. These improvements stem from the model’s self-reflection capability, which identifies the most reliable analyses among multiple outputs to derive a more robust final prediction. Importantly, the gain is more pronounced for the smaller model: the 3.7-point improvement for Qwen3-4B is substantially larger than the 2.2-point gain for the 8B version. We hypothesize that this is because smaller-scale models exhibit more obvious analytical defects, which the self-reflection mechanism is particularly effective at identifying. Consequently, our enhanced smaller model even surpasses the larger baseline: System 10 attains an average accuracy of 76.4, outperforming System 12’s score of 76.1.

5 Analysis

This section aims to answer the following research questions: What are the optimal configurations for training and inference in ReflectRM? (5.1) How do the dual capabilities within our unified framework mutually reinforce each other? (5.2) How robust and scalable is ReflectRM across different settings? (5.3) Unless otherwise specified, all experiments are conducted using Qwen3-4B, with evaluations on RewardBench (abbreviated as R.B.) and RMB.

5.1 Optimal Configurations of ReflectRM

Optimal Data Mixing Ratio While ReflectRM demonstrates strong performance in pairwise response preference judgment, identifying the optimal mixing ratio between **Pref.** and **Refl.** data remains a critical factor for our framework. We train several variants with ratios ranging from 1:0 (containing only Pref. data) to 5:1 (Pref. to Refl. data), all of which are evaluated using standard

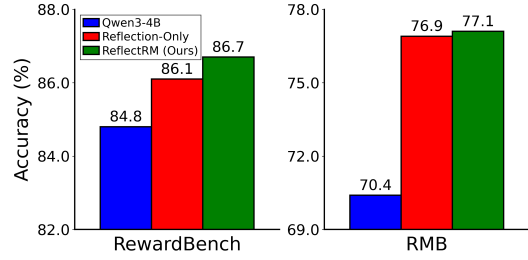


Figure 4: Impact of preference data on self-reflection. Preference data provides a robust foundation for the evaluative logic used to judge analytical processes.

System	Qwen3-4B	Qwen3-8B	Qwen3-14B
Base Model	72.7	75.1	75.7
ReflectRM (Ours)	76.4	77.3	79.5

Table 4: Performance scaling across model sizes. ReflectRM yields consistent performance gains, demonstrating the framework’s high scalability.

greedy decoding. As illustrated in Figure 3, the model’s performance peaks at a 4:1 ratio. Interestingly, an excessive amount of reflection data (e.g., 2:1) leads to a slight decline in performance. We hypothesize that an excessive proportion of reflection tasks may distract the model from its primary objective of response preference modeling. These results confirm that a 4:1 ratio strikes an optimal balance for the unified training process.

Effectiveness of Inference Strategy To investigate the contribution of each component within our two-stage strategy, we evaluate its two key stages: confidence-guided anchor selection and self-reflection filtering. Specifically, we compare the full ReflectRM against two variants: (1) **Random Anchor**, which selects the anchor output randomly rather than via confidence scores; and (2) **Random Winners**, which replaces the self-reflection process with random sampling, selecting a number of outputs equal to the size of ReflectRM’s original winner group.

As shown in Table 2, both variants lead to performance degradation. Notably, the drop is significantly more pronounced when the self-reflection stage is bypassed, demonstrating that the model’s capability to evaluate its own analysis is the primary driver of inference-time performance gains. This validates that combining a high-confidence anchor with self-reflection filtering is essential for the effectiveness of ReflectRM. Besides, we provide a more comprehensive ablation in Appendix A.2, including comparisons with simpler test-time strate-

ID	System	JudgeBench	RewardBench	RM-Bench	RMB	PPE-Preference	Overall	
							(1k)	(1k)
7	Qwen3-4B	49.3	74.8	60.5	52.3	34.9	54.3	-
8	4 + RFT	53.1	75.1	67.1	56.5	46.6	59.7	5.4
9	4 + Unified Training	53.1	78.5	64.3	64.3	48.1	61.7	7.4
10	6 + Two-Stage Inference (ReflectRM)	54.9	81.1	66.6	68.3	51.7	64.5	10.2
11	Qwen3-8B	52.7	78.4	64.9	60.9	42.1	59.8	-
12	8 + RFT	48.9	78.8	65.6	67.0	44.6	61.0	1.2
13	8 + Unified Training	53.9	83.6	72.6	58.4	44.2	62.5	2.7
14	10 + Two-Stage Inference (ReflectRM)	56.3	84.9	73.5	59.4	44.3	63.7	3.9

Table 5: Evaluation of positional consistency across five benchmarks. Qwen3-4B and Qwen3-8B are chosen as the backbone model. The best results under each backbone are labeled using **bold font**. ReflectRM achieves a substantial improvement of up to +10.2 points, nearly doubling the consistency gains of the standard RFT baseline, showing that supervising the analytical trace leads to a more stable and reliable judgment process.

gies (e.g., Review & Revise) and evaluations under matched computational budgets to isolate the source of our performance gains.

5.2 Mutual Reinforcement in ReflectRM

Impact of Refl. on Response Preference We explore how reflection data benefits the model’s core response preference capability. Specifically, we establish an initial baseline by training a model solely on the 13.7k Pref. data. To account for the influence of total training volume, we further construct a scaled variant using 17.1k samples of pure Pref. data. Finally, to ensure a fair comparison, all models in this experiment are evaluated using standard greedy decoding without utilizing the self-reflection capability at inference time.

As shown in Table 3, the model trained on mixed data significantly outperforms the variant trained on an equal volume of pure preference data. This result demonstrates that the reflection data provides a more effective learning signal, enhancing the model’s judgment capability more efficiently than simply scaling preference data alone. By explicitly supervising the analytical trace, the model internalizes a more robust underlying logic, which directly improves its response preference performance even without inference-time enhancements. Notably, this performance gain is consistent across different model capacities. Extended experiments demonstrating the scalability of our mixed-data strategy are included in Appendix A.4.

Impact of Pref. on Analysis Preference Additionally, we investigate whether learning the response preference task provides an essential foundation for the self-reflection capability. To isolate this effect, we compare ReflectRM against a

variant employing an independent reflection model (**Reflection-Only**, trained exclusively on Refl.) to perform the Inference Stage 2 described in Section 3.3. Crucially, both methods use the same initial outputs generated in Stage 1, ensuring the performance difference is driven solely by the quality of the self-reflection step.

As illustrated in Figure 4, ReflectRM consistently outperforms Reflection-Only. This gap illustrates that preference data also plays a vital role in enhancing the model’s self-reflection capability. Within our unified framework, the ability to evaluate analytical processes is grounded in the core preference judgment ability learned from the preference data. As a result, ReflectRM can identify high-quality analytical traces with greater accuracy than a model trained only on reflection data. This provides strong evidence that the dual capabilities of ReflectRM are mutually reinforcing manifestations of the same underlying judgment ability, rather than a mere aggregation of disparate tasks.

5.3 Robustness and Scalability of ReflectRM

To thoroughly investigate the robustness and scalability of ReflectRM, we conduct evaluations across all benchmarks. Specifically, scalability is reported based on four benchmarks: RewardBench, RM-Bench, RMB, and PPE-Preference. Positional bias is analyzed using the complete JudgeBench dataset and 1,000 randomly sampled instances from each of the other four benchmarks.

Scalability across Model Sizes We examine whether the performance gains of ReflectRM remain consistent as model capacity increases. We evaluate ReflectRM across 4B, 8B, and 14B parameter scales. As shown in Table 4, our method con-

sistently yields improvements for all model sizes. These results demonstrate that ReflectRM is a scalable framework that provides a consistent boost independent of model capacity.

Robustness to Positional Bias Positional bias, the tendency for reward models to favor a response based on its presentation order (e.g., always preferring the first candidate) rather than its content, is a major challenge that leads to inconsistent and unstable judgments. In this part, we evaluate the effectiveness of ReflectRM in mitigating this bias by measuring positional consistency, where a sample is considered correct only if the model identifies the preferred response in **both** possible orderings. Specifically, we conduct thorough tests across five benchmarks while maintaining the same training configurations described in Section 4.

As shown in Table 5, ReflectRM shows a significant improvement in positional consistency. Our method achieves a **10.2-point gain** over the base model, which is nearly double the 5.4-point gain of the RFT baseline. Notably, this improvement in consistency (10.2) is far more substantial than the gain in standard accuracy (3.7) reported in Table 1. This disparity highlights the unique advantage of ReflectRM: by critiquing its own analytical process, the model learns to align its final decision with the analytical logic rather than positional orderings. This makes ReflectRM not only more accurate but also a far more reliable evaluator, demonstrating its significant potential to mitigate the long-standing challenge of positional bias in reward modeling.

6 Conclusion

We introduced **ReflectRM**, a framework that unifies response preference and process evaluation into a single generative objective. Our results demonstrate that these two capabilities are mutually reinforcing, enabling the model to internalize a more consistent and robust evaluative logic. By leveraging a two-stage inference strategy, ReflectRM significantly outperforms standard baselines across diverse benchmarks. Most notably, our method achieves a substantial improvement in positional consistency, demonstrating that supervising the analytical trace effectively mitigates positional bias and leads to more reliable reward modeling.

Limitations

A primary limitation of ReflectRM is the increased computational overhead associated with its two-

stage inference strategy. Compared to conventional inference-time scaling methods that perform majority voting across N independent rollouts, our approach effectively doubles the computational requirements, as the self-reflection stage necessitates $N - 1$ additional pairwise comparisons to filter the outputs. However, it’s worth noting that this extra computational cost can be mitigated through acceleration methods, such as quantification and speculative decoding.

Ethics Statement

This work follows the ACL Ethics Policy. Our findings are based on publicly available datasets for reproducibility purposes. We acknowledge that LLMs may exhibit inherent societal biases and are prone to hallucinations. Therefore, if someone finds our work interesting and would like to use it in a specific environment, we strongly suggest the user conduct safety and bias evaluations to mitigate potential risks.

Acknowledgments

This work was supported by the Shenzhen Science and Technology Program (Grant No. RCJC20210706091946001) and the Shenzhen Science and Technology Program (Grant No. ZDCY20250901104207008). We would like to thank the anonymous reviewers and meta-reviewer for their insightful suggestions.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Nuo Chen, Zhiyuan Hu, Qingyun Zou, Jiaying Wu, Qian Wang, Bryan Hooi, and Bingsheng He. 2025a. Judgelrm: Large reasoning models as a judge. *arXiv preprint arXiv:2504.00050*.
- Xiuxi Chen, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru Wang, Yu Zhang, Denghui Zhang, Tong Zhang, and 1 others. 2025b. Rmr1: Reward modeling as reasoning. *arXiv preprint arXiv:2505.02387*.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Maric, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

- Jacob Dineen, Aswin RRV, Qin Liu, Zhikun Xu, Xiao Ye, Ming Shen, Zhaonan Li, Shijie Lu, Chitta Baral, Muhao Chen, and Ben Zhou. 2025. Qa-lign: Aligning llms through constitutionally decomposed qa. *arXiv preprint arXiv:2506.08123*.
- Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. 2024. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*.
- Ahmed El-Kishky, Alexander Wei, Andre Saraiva, Borys Minaiev, Daniel Selsam, David Dohan, Francis Song, Hunter Lightman, Ignasi Clavera, Jakub Pachocki, and 1 others. 2025. Competitive programming with large reasoning models. *arXiv preprint arXiv:2502.06807*.
- Evan Frick, Tianle Li, Connor Chen, Wei-Lin Chiang, Anastasios N Angelopoulos, Jiantao Jiao, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024. How to evaluate reward models for rlhf. *arXiv preprint arXiv:2410.14872*.
- Yichao Fu, Xuwei Wang, Yuandong Tian, and Jiawei Zhao. 2025. Deep think with confidence. *arXiv preprint arXiv:2508.15260*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. 2025. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025a. Deepseek-rl: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Jiaxin Guo, Zewen Chi, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. 2025b. [Reward reasoning models](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.
- Nathan Lambert. 2025. Reinforcement learning from human feedback. *arXiv preprint arXiv:2504.12501*.
- Nathan Lambert, Valentina Pyatkin, Jacob Morrison, Lester James Validad Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, and 1 others. 2025. Rewardbench: Evaluating reward models for language modeling. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1755–1797.
- Derek Li, Jiaming Zhou, Amirreza Kazemi, Qianyi Sun, Abbas Ghaddar, Liheng Ma, Yu Luo, Dong Li, Jianye HAO, and Yingxue Zhang. 2025. Omnithinker: Scaling cross-domain generalization in llms via multi-task rl with hybrid rewards. In *2nd AI for Math Workshop@ ICML 2025*.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- Chris Yuhao Liu, Liang Zeng, Jiakai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. 2024b. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*.
- Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiakai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, and 1 others. 2025a. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *arXiv preprint arXiv:2507.01352*.
- Fengyuan Liu, Nouar AlDahoul, Gregory Eady, Yasir Zaki, and Talal Rahwan. 2024c. Self-reflection makes large language models safer, less biased, and ideologically neutral. *arXiv preprint arXiv:2406.10400*.
- Liangxin Liu, Xuebo Liu, Derek F Wong, Dongfang Li, Ziyi Wang, Baotian Hu, and Min Zhang. 2024d. Selectit: Selective instruction tuning for llms via uncertainty-aware self-reflection. *Advances in Neural Information Processing Systems*, 37:97800–97825.
- Shudong Liu, Hongwei Liu, Junnan Liu, Linchen Xiao, Songyang Gao, Chengqi Lyu, Yuzhe Gu, Wenwei Zhang, Derek F Wong, Songyang Zhang, and 1 others. 2025b. Compassverifier: A unified and robust verifier for llms evaluation and outcome reward. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33454–33482.
- Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. 2024e. Rm-bench: Benchmarking reward models of language models with subtlety and style. *arXiv preprint arXiv:2410.16184*.
- Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. 2025c. Inference-time scaling for generalist reward modeling. *arXiv preprint arXiv:2504.02495*.
- Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zeyun Ma, and Wenhui Chen. 2025. General-reasoner: Advancing llm reasoning across all domains. *arXiv preprint arXiv:2505.14652*.

- Dakota Mahan, Duy Van Phung, Rafael Rafailov, Chase Blagden, Nathan Lile, Louis Castricato, Jan-Philipp Fränken, Chelsea Finn, and Alon Albalak. 2024. Generative reward models. *arXiv preprint arXiv:2410.12832*.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Junsoo Park, Seungyeon Jwa, Ren Meiyang, Daeyoung Kim, and Sanghyuk Choi. 2024. Offsetbias: Leveraging debiased data for tuning evaluators. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1043–1067.
- Hoang Phan, Victor Li, and Qi Lei. 2025. Think twice, generate once: Safeguarding by progressive self-reflection. *arXiv preprint arXiv:2510.01270*.
- Matthew Renze and Erhan Guven. 2024. Self-reflection in llm agents: Effects on problem-solving performance. *arXiv preprint arXiv:2405.06682*.
- ByteDance Seed, Jiase Chen, Tiantian Fan, Xin Liu, Lingjun Liu, Zhiqi Lin, Mingxuan Wang, Chengyi Wang, Xiangpeng Wei, Wenyuan Xu, and 1 others. 2025. Seed1. 5-thinking: Advancing superb reasoning models with reinforcement learning. *arXiv preprint arXiv:2504.13914*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021.
- Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Y Tang, Alejandro Cuadron, Chenguang Wang, Raluca Ada Popa, and Ion Stoica. 2024. Judgebench: A benchmark for evaluating llm-based judges. *arXiv preprint arXiv:2410.12784*.
- Kimi Team, Angang Du, Bofei Gao, Bofei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.
- Chenglong Wang, Yang Gan, Yifu Huo, Yongyu Mu, Qiaozhi He, Murun Yang, Bei Li, Tong Xiao, Chunliang Zhang, Tongran Liu, and 1 others. 2025a. Gram: A generative foundation reward model for reward generalization. *arXiv preprint arXiv:2506.14175*.
- Chenglong Wang, Yongyu Mu, Hang Zhou, Yifu Huo, Ziming Zhu, Jiali Zeng, Murun Yang, Bei Li, Xiaoyang Hao, Chunliang Zhang, and 1 others. 2026. Gram-r²: Self-training generative foundation reward models for reward reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 33395–33403.
- Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024a. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *arXiv preprint arXiv:2406.12845*.
- Tianlu Wang, Iliia Kulikov, Olga Golovneva, Ping Yu, Weizhe Yuan, Jane Dwivedi-Yu, Richard Yuanzhe Pang, Maryam Fazel-Zarandi, Jason Weston, and Xian Li. 2024b. Self-taught evaluators. *arXiv preprint arXiv:2408.02666*.
- Yutong Wang, Jiali Zeng, Xuebo Liu, Fandong Meng, Jie Zhou, and Min Zhang. 2024c. Taste: Teaching large language models to translate through self-reflection. *arXiv preprint arXiv:2406.08434*.
- Zhilin Wang, Jiaqi Zeng, Olivier Delalleau, Hoo-Chang Shin, Felipe Soares, Alexander Bukharin, Ellie Evans, Yi Dong, and Oleksii Kuchaiev. 2025b. Helpsteer3-preference: Open human-annotated preference data across diverse tasks and languages. *arXiv preprint arXiv:2505.11475*.
- Chenxi Whitehouse, Tianlu Wang, Ping Yu, Xian Li, Jason Weston, Iliia Kulikov, and Swarnadeep Saha. 2025. J1: Incentivizing thinking in llm-as-a-judge via reinforcement learning. *arXiv preprint arXiv:2505.10320*.
- Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason E Weston, and Sainbayar Sukhbaatar. 2025. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 11548–11565.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Ziyi Ye, Xiangsheng Li, Qiuchi Li, Qingyao Ai, Yujia Zhou, Wei Shen, Dong Yan, and Yiqun Liu. 2025. Learning llm-as-a-judge for preference alignment. In *The Thirteenth International Conference on Learning Representations*.
- Jiachen Yu, Shaoning Sun, Xiaohui Hu, Jiayu Yan, Kaidong Yu, and Xuelong Li. 2025. Improve llm-as-a-judge ability as a general ability. *arXiv preprint arXiv:2502.11689*.
- Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2024. Generative verifiers: Reward modeling as next-token prediction. *arXiv preprint arXiv:2408.15240*.

Enyu Zhou, Guodong Zheng, Binghai Wang, Zhiheng Xi, Shihan Dou, Rong Bao, Wei Shen, Limao Xiong, Jessica Fan, Yurong Mou, and 1 others. 2024. Rmb: Comprehensively benchmarking reward models in llm alignment. *arXiv preprint arXiv:2410.09893*.

A Appendix

A.1 Details of Training Data Construction

The construction of the ReflectRM training dataset follows a two-step pipeline designed to generate high-quality supervision signals for both response preference modeling and process-level self-reflection.

Pref. Data For each query in the HelpSteer3 dataset, we generate eight independent outputs using the instruction-tuned base model at a sampling temperature of 1.0. Each output consists of a textual analysis a and a prediction p indicating the preferred response. This sampling-based approach provides a diverse set of analytical paths for each problem. To focus on informative samples, we exclude easy instances where the model’s prediction is correct across all eight trials, as these offer limited signals for further optimization. From the remaining pool, we randomly sample approximately 13.7k instances to form the **Pref.** data.

Ref. Data To construct the reflection data, we leverage the subset of queries that yielded mixed outcomes (i.e., both correct and incorrect predictions) across the eight rollouts. For each such query, we pair the analysis from a correct output a_{cor} with the analysis from an incorrect one a_{inc} in a random order. Following the principle that the reliability of a GRM’s final judgment is intrinsically determined by the quality of its analytical process, which is similar to the Chain-of-Thought (CoT) process in reasoning models, we designate (a_{cor}) as the preferred analysis. To maintain data diversity and balance, we generate exactly one reflection pair per query that exhibited inconsistent predictions. This automated pipeline allows us to synthesize thousands of analysis preference pairs without the need for manual annotation.

Final Training Dataset As summarized in Table 6, the final training set maintains a 4:1 ratio between standard preference data and reflection data. This mixture ratio was selected based on the empirical results in Section 5.1, ensuring that the model maintains its primary capability in response preference judgment while effectively internalizing the self-reflection capability.

Backbone	Pref.	Ref.	Sum
Qwen3-4B	13,692	3,420	17,112
Qwen3-8B	13,560	3,380	16,940
Qwen3-14B	13,534	3,384	16,918

Table 6: Detailed statistics of the training dataset.

System	R.B.	RMB	AVG	Δ
ReflectRM	86.7	77.1	81.9	-
w/ Anchor-only	81.1	74.5	77.8	-4.1
Review & Revise	84.5	71.5	78.0	-3.9

Table 7: Comparison with simpler inference strategies.

A.2 Additional Ablations on Inference Mechanism

In this section, we provide a deeper investigation into the effectiveness of ReflectRM’s inference mechanism from two perspectives: strategy complexity and computational cost.

A.2.1 Comparison with Simpler Test-time Strategies

We further isolate the contribution of the self-reflection process by comparing ReflectRM against two simpler variants: (1) **Anchor-only**: Simply selecting the rollout with the highest confidence score as the final decision, bypassing the self-reflection mechanism; and (2) **Review & Revise**: A standard two-pass refinement strategy where the model critiques and updates its initial judgment.

The results in Table 7 show that ReflectRM outperforms both baselines. The gap between ReflectRM and "Anchor-only" confirms that while a high-confidence anchor is a strong starting point, the aggregation of multiple reliable analytical traces via reflection is crucial for robust judgment. Furthermore, the superiority over "Review & Revise" suggests that our performance gains stem from the specialized self-reflection process rather than a mere second-chance mechanism.

A.2.2 Performance under Matched Computational Budgets

A potential concern is whether the gains of ReflectRM stem simply from increased computation during inference. To ensure a fair comparison, we evaluate a baseline GRM using Majority Voting with $2N - 1 = 15$ rollouts, which matches the total computational budget of ReflectRM ($N = 8$ initial rollouts plus $N - 1 = 7$ pairwise reflection comparisons).

System	RewardBench	RM-Bench	RMB	PPE-Preference	AVG	Δ
GRM	84.0	76.5	71.3	62.7	73.6	-
ReflectRM (Ours)	86.7	77.7	77.1	64.2	76.4	+2.8

Table 8: Performance comparison under matched computational budgets.

System	RewardBench	RM-Bench	RMB	PPE-Preference	AVG	Δ
Qwen3-4B	84.8	75.4	70.4	60.1	72.7	-
+ RFT	83.1	75.1	74.1	62.2	73.6	+0.9
+ ReflectRM (Ours)	86.7	76.1	72.5	64.1	74.9	+2.2

Table 9: Main Performance on Skywork-Reward-V2 with Qwen3-4B.

System	JudgeBench	RewardBench (1k)	RM-Bench (1k)	RMB (1k)	PPE-Preference (1k)	AVG	Δ
Qwen3-4B	49.3	74.8	60.5	52.3	34.9	54.3	-
+ RFT	54.6	77.6	67.5	57.5	36.2	58.7	+4.4
+ ReflectRM (Ours)	57.6	79.6	69.1	59.3	40.8	61.3	+7.0

Table 10: Positional Consistency on Skywork-Reward-V2 with Qwen3-4B.

System	R.B.	RMB	AVG
Qwen3-4B	84.8	70.4	77.6
+ Pref.	84.0	71.2	77.6
+ Pref. & Refl.	85.3	75.3	80.3
Qwen3-8B	85.8	72.9	79.4
+ Pref.	85.0	73.9	79.5
+ Pref. & Refl.	87.9	73.0	80.5
Llama-3.1-8B-Instruct	74.8	62.7	68.8
+ Pref.	75.6	63.9	69.8
+ Pref. & Refl.	76.6	65.9	71.3

Table 11: Performance on RewardBench (R.B.) and RMB with mixed training data across different scales and architectures.

As shown in Table 8, even when the baseline is granted an equivalent computational budget, ReflectRM maintains a significant performance lead. This demonstrates that the gain of our method is derived from the structured evaluative logic of the self-reflection voting mechanism rather than a mere increase in inference-time compute.

A.3 Robustness Across Different Training Distributions

To demonstrate the robustness of ReflectRM across different training distributions, we train Qwen3-4B on Skywork-Reward-V2 dataset (Liu et al., 2025a) with the same 4:1 mixture ratio.

As shown in Table 9 and Table 10, ReflectRM yields consistent improvements in both overall accuracy and positional consistency compared to the RFT baseline. These results confirm that

the benefits of process-level supervision are not dataset-specific but generalize effectively to other preference-training distributions.

A.4 Scalability and Cross-Architecture Robustness

We investigate whether the performance gains of ReflectRM scale with model capacity and generalize across architectures. We extend the experiments in Table 3 to include Qwen3-8B and Llama-3.1-8B-Instruct (Grattafiori et al., 2024).

As summarized in Table 11, the improvement remains consistent across different model sizes and families, confirming that our unified judgment framework is a scalable and robust solution for generative reward modeling.

A.5 Prompt Templates

Following our unified framework described in Section 3.1, we implement a standardized prompting strategy to accommodate response preference and analysis preference tasks. As illustrated in Figure 5, the templates are designed to align with the components of Equation (1):

- **Condition ϕ :** This is represented by the Context field. For response preference, ϕ includes the conversation history and the user’s query q . For analysis preference, ϕ is expanded to include the two candidate responses (r_1, r_2) to provide the necessary context for evaluating the subsequent analyses (i.e. critiques).

- **Candidates** δ : This part corresponds to the Responses for Judgment or Critiques for Judgment fields, containing the two entities to be compared.
- **Output** (a, p) : We mandate a structured output format using `<Analysis>` and `<Result>` tags. The content within `<Analysis>` corresponds to the textual analysis a . The `<Result>` tag contains a preference statement (e.g., Response 1 is better than Response 2), which is mapped to the preference index $p \in \{1, 2\}$.

This case clearly illustrates how the self-reflection capability allows ReflectRM to derive a more robust and reliable result.

Notably, we append the `</no_think>` token at the end of each prompt. This is a model-specific control token for the Qwen3 backbone that deactivates its internal reasoning mode. This prompting design ensures strict alignment with our unified generative formulation while maintaining clear task distinction, effectively preventing the model from confusing the two preference capabilities.

A.6 Case Study

We present a case study of ReflectRM’s two-stage inference pipeline in Table 12. In this scenario, the user provides a specific instruction to keep responses "succinct and to the point". Notably, the original query from PPE-Preference contains minor errors (e.g., 'succint the to the point'). These are preserved verbatim to ensure the case study faithfully reflects the source benchmark. Both responses are concise, but Response 2 includes a proactive follow-up question.

During the first stage of inference, the model generates $N = 8$ outputs to judge the preference between two responses. We select two representative outputs to illustrate the process (including the anchor output). Interestingly, the selected anchor over-emphasizes the literal interpretation of the brevity constraint, penalizing Response 2’s follow-up as "not necessary". Consequently, this high-confidence anchor initially leads to an incorrect preference prediction.

In the second stage, ReflectRM leverages its self-reflection capability to evaluate the analytical quality of outputs against the anchor. As shown in the Self-Reflection section of Table 12, the model identifies the analytical oversight in Critique 1, recognizing that Critique 2 provides a more comprehensive and context-aware evaluation, and finally derives a correct prediction.

Prompt of Response Preference

You are a discerning and impartial Judge. In the context of the conversation provided below (with the user's query being the last round), your role is to evaluate the quality of the two 'Responses' and determine which is better. Your decision should be based on which response better aligns with the user's instructions and more effectively addresses their query. To render a fair judgment, you need to think step-by-step to conduct a deep analysis, clearly articulating the reasoning for your decision. Your judgment must be free of any positional or length biases.

Context

[The Begin of Conversation Context & Query]\n<context>\n[The End of Conversation Context & Query]

Responses for Judgment

These are the two responses you must analyze and compare.

[The Begin of Response 1]\n<response 1>\n[The End of Response 1]
[The Begin of Response 2]\n<response 2>\n[The End of Response 2]

Your Structured Judgment

Follow these steps precisely and use the specified tags for your output. 1. Provide Detailed Analysis: Think step-by-step to conduct a detailed analysis of the Responses. Place your analysis within `<Analysis>` tags. 2. Render Final Verdict: Conclude with your final verdict based on your analysis. State which response is better in the format "Response 1 is better than Response 2" or "Response 2 is better than Response 1". Place this final choice within `<Result>` tags. Your entire output must follow the format below. <Analysis> Your detailed step-by-step analysis of the two responses. </Analysis> \n <Result> \n Based on your Analysis, only print the following: "Response 1 is better than Response 2" OR "Response 2 is better than Response 1". </Result> /no_think

Prompt of Analysis Preference

You are a discerning and impartial Judge. Your role is to evaluate the quality of the two 'Critiques' presented below. These critiques are themselves analyses of two original 'Responses' generated for a conversation (with the user's query being the last round). Your decision should be based on which critique provides a more insightful, accurate, fair, and well-reasoned analysis. To render a fair judgment, you need to think step-by-step to conduct a deep analysis, clearly articulating the reasoning for your decision. Your judgment must be free of any positional or length biases.

Context

[The Begin of Conversation Context & Query]\n<context>\n[The End of Conversation Context & Query]
[The Begin of Response 1]\n<response 1>\n[The End of Response 1]
[The Begin of Response 2]\n<response 2>\n[The End of Response 2]

Critiques for Judgment

These are the two critiques you must analyze and compare. Each one analyzes 'Response 1' and 'Response 2' shown in 'Context'.

[The Begin of Critique 1]\n<critique 1>\n[The End of Critique 1]
[The Begin of Critique 2]\n<critique 2>\n[The End of Critique 2]

Your Structured Judgment

Follow these steps precisely and use the specified tags for your output. 1. Provide Detailed Analysis: Think step-by-step to conduct a detailed analysis of the Critiques. Place your analysis within `<Analysis>` tags. 2. Render Final Verdict: Conclude with your final verdict based on your analysis. State which critique is better in the format "Critique 1 is better than Critique 2" or "Critique 2 is better than Critique 1". Place this final choice within `<Result>` tags. Your entire output must follow the format below. <Analysis> Your detailed step-by-step analysis of the two critiques. </Analysis> \n <Result> \n Based on your Analysis, only print the following: "Critique 1 is better than Critique 2" OR "Critique 2 is better than Critique 1". </Result> /no_think

Figure 5: Prompt templates for the unified judgment framework. The top template is used for pairwise response preference judgment, while the bottom one is for pairwise analysis preference (self-reflection).

Query and Responses	
Query	In this conversation, please keep your responses succinct the to the point. Please avoid lengthy explanations or examples unless instructed.
Response 1	Understood! I'll keep my responses brief and to the point.
Response 2	Absolutely, I'll keep my responses concise and to the point. How can I assist you today?
Output 1	
Analysis	<p>The user requested that responses be “succinct the to the point” and to avoid lengthy explanations or examples unless instructed. Response 1 simply says “Understood! I'll keep my responses brief and to the point.” while Response 2 adds “Absolutely, I'll keep my responses concise and to the point. How can I assist you today?”</p> <p>Response 2 is more comprehensive as it not only confirms understanding of the user’s instruction but also actively asks how the user wants to proceed, which shows engagement and willingness to assist. However, the user’s query was simply to confirm that the responses would be brief and to the point, and the initial response from the assistant was already meeting that requirement. Response 1 is more in line with the user’s explicit instruction to be “succinct the to the point” as it directly confirms understanding without adding extra information. Response 2, while polite, includes an additional question that may not be necessary if the user’s query was solely about confirming the response style. Critique 1 Anchor</p>
Predict	<i>Response 1 is better than Response 2</i> ✗
Output 2	
Analysis	<p>The user instructed to keep responses succinct and to the point, avoiding lengthy explanations or examples unless instructed. Response 1 simply says “Understood! I'll keep my responses brief and to the point.” This directly acknowledges the instruction but does not offer any further assistance or information. Response 2, however, not only acknowledges the instruction but also actively asks how the user wants to proceed, showing a willingness to assist and engage in the conversation. While both responses are concise, Response 2 provides a more effective alignment with the user’s instructions by demonstrating proactive engagement and offering further assistance, which is more in line with the context of a conversational interaction. Critique 2</p>
Predict	<i>Response 2 is better than Response 1</i> ✓
Self-Reflection	
Analysis	<p>The two critiques provide different perspectives on the analysis of Response 1 and Response 2. Critique 1 focuses on the difference between the two responses by highlighting that Response 2 includes an additional question that may not be necessary if the user’s query was solely about confirming the response style. It argues that Response 1 is more in line with the user’s explicit instruction because it directly confirms understanding without adding extra information. However, Critique 2 emphasizes that Response 2 provides a more effective alignment with the user’s instructions by demonstrating proactive engagement and offering further assistance, which is more in line with the context of a conversational interaction.</p> <p>Critique 2 offers a more comprehensive analysis by recognizing that while both responses are concise, Response 2 demonstrates proactive engagement and offers further assistance, which is crucial in a conversational context. It also acknowledges the user’s instruction to keep responses succinct but emphasizes the importance of active engagement in a conversation. In contrast, Critique 1 is slightly more focused on the literal interpretation of the user’s instruction and may overlook the conversational context by suggesting that the additional question in Response 2 is unnecessary.</p>
Predict	<i>Critique 2 is better than Critique 1</i>
Final Predict	Response 2 is better than Response 1 ✓

Table 12: A case study of ReflectRM’s two-stage inference strategy. By reflecting on the analytical processes, ReflectRM finally derives the correct verdict.