

# Generative Reward Modeling via Synthetic Criteria Preference Learning

Xiaobo Liang<sup>♣, ♥</sup>, Haoke Zhang<sup>♣, ♥</sup>, Juntao Li<sup>♣, ♥\*</sup>

Kehai Chen<sup>♦</sup>, Qiaoming Zhu<sup>♣, ♥</sup>, Min Zhang<sup>♣, ♥</sup>

<sup>♣</sup> Soochow University, <sup>♦</sup> Harbin Institute of Technology, Shenzhen, China

<sup>♥</sup> Key Laboratory of Data Intelligence and Advanced Computing, Soochow University

{xbliang, ljt, qmzhu, minzhang}@suda.edu.cn

hkzhangnlp@stu.suda.edu.cn chenkehai@hit.edu.cn

## Abstract

Generative Reward Models (GenRMs) leverage synthesized Chains of Thought (CoT) to reduce the need for massive labeled data, but this approach introduces risks of overoptimization due to the inability to guarantee the correctness of the CoTs. Identifying and optimizing unexpected behaviors within these synthesized CoT remains a challenge, as it heavily depends on precise annotations of intermediate behavior, similar to process supervision. In this work, we introduce a criteria-based preference tree for reward modeling, where each path in the tree represents a reasoning trajectory based on synthesized criteria. Crucially, each reasoning trajectory can be independently optimized through RL algorithm. These fine-grained process reward signals are derived from the inference-time computations and predefined rules, eliminating the need for human supervision. In experiments, **SyncPL**<sup>1</sup> showed significant improvements over baselines on multiple human preference benchmarks. We further demonstrate that synthesized data can be learned using a long CoT format, analogous to an o1-like model, further enhancing performance while keeping stability and efficiency during training.

## 1 Introduction

With Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Schulman et al., 2017; Ouyang et al., 2022), large language models (LLMs) can effectively follow user instructions and generate responses that align with human values. Traditionally, discriminative reward models (e.g., Bradley-Terry models (Bradley and Terry, 1952)) have been used as proxies to evaluate human preferences. These reward models **lack explicit reasoning**, leading to significant *overoptimization* issues (Skalse et al., 2022; Chen et al., 2024) and often deteriorating on out-of-distribution data (Zhang

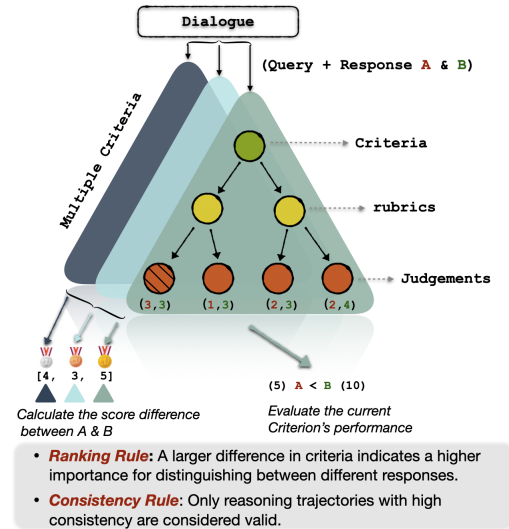


Figure 1: Illustration of SyncPL.

et al., 2024). The reasoning models like OpenAI o1 and DeepSeek R1 (Guo et al., 2025) demonstrates that powerful CoT reasoning capability is key to performance improvement and scalability. As a potential alternative, GenRMs (Mahan et al., 2024; Ye et al., 2024) leverage the capabilities of LLMs, such as CoT reasoning and inference-time scaling, offering an interpretable evaluation process. However, this approach does not guarantee that the model generates reasonable CoT trajectory during inference. Especially in open-domain scenarios, the model often needs to consider multiple evaluation criteria (Liu et al., 2023; Wang et al., 2024b), such as accuracy, fluency, and consistency.

Recent studies (Kim et al., 2023; Saha et al., 2023; Liang et al., 2024) have used LLMs to synthesize criteria-based CoT for responses evaluation. In Section 3, we demonstrate how these methods can address challenges in reward modeling, such as implicit modeling, thus improving the accuracy of preference judgments. Additionally, GenRM ensures the effective parallel generation of multiple criteria to achieve more accurate judgment results. However, the **compounding error of synthetic**

\* Corresponding Author

<sup>1</sup>SyncPL means Synthetic Criteria Preference Learning.

**CoT** hinders the model’s improvement through self-improvement, as the model struggles to verify the correctness of the CoT trajectories. To address this issue, we predefined two rules as meta criteria to filter high-quality trajectories.

As shown in Figure 1, the CoT trajectories in our proposed method, **SyncPL**, are composed of three essential elements: *criteria*, *rubrics*, and *judgments*. The *judgments* includes scores assigned to different responses (e.g., A and B), such as (1, 3). Furthermore, we will utilize the Monte Carlo rollout policy to expand CoT trajectories into a preference tree, which provides a manner to estimate the expected reward of different trajectories. To identify and optimize critical *criteria*, our method contains two stages: In the ranking phase, we first compute the score differences between responses A and B across various reasoning trajectories. By aggregating these differences, we determine the importance of each criterion. Specifically, the **Ranking Rule** identifies the relative importance of the *criteria*, while the **Consistency Rule** filters out trajectories with low confidence and inconsistent judgments. In the learning phase, we utilize these ranked trajectories to optimize LLMs through Rejection Sampling Fine-tuning (RSFT) and Direct Preference Optimization (Rafailov et al., 2024) (DPO). **SyncPL** achieves substantial performance gains across multiple preference benchmarks both on in-distribution and out-of-distribution (OOD) scenarios.

Benefiting from the rigorous data filtering, we attempted to combine different criteria-based judgments into a long CoT format and conducted RL training. We demonstrate that the SyncPL-o1-8 works effectively, surpassing the performance of the LLaMA3.1-70B. In a nutshell, our contributions are as follows:

- We are the first to introduce tree-based preference learning into generative reward modeling to optimize synthetic criteria trajectories.
- SyncPL-DPO prioritizes key and high-quality criteria, effectively reducing inference costs.
- We are the first to demonstrate and implement the o1-like long CoT format<sup>2</sup> for GenRM through criteria-based process supervision.

<sup>2</sup>It denotes multi-step reasoning across different criteria rather than merely a long response length.

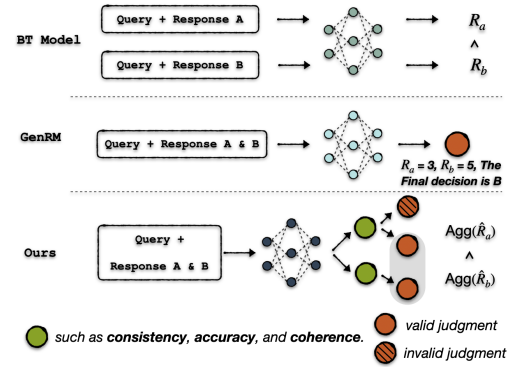


Figure 2: The difference between our proposed SyncPL and existing reward modeling approaches.

## 2 Related Works

Reward modeling plays a crucial role in aligning LLMs with human values and behaviors (Ouyang et al., 2022; Bai et al., 2022), particularly in ensuring their safe and effective deployment. To reduce the cost of annotating preference data, reward models are often used as proxies to provide reward signals for the policy model (Rafailov et al., 2024; Azar et al., 2024). Given the user query and the candidate responses A and B, the two main approaches are discriminative RM and generative RM.

**Discriminative RM.** The Bradley-Terry (Bradley and Terry, 1952) model improves pairwise preference selection through Luce-Shephard Choice Rule (Luce, 1959; Shepard, 1957). However, this method is prone to reward hacking (Yang et al., 2024; Coste et al., 2023; Pace et al., 2024; Shen et al., 2024), also known as *overoptimization*, as it easily learns spurious correlations between data and preference label. For example, Chen et al. (2024) demonstrate that verbose but less helpful responses often receive higher reward scores. To address this issue, Shen et al. (2023) introduce consistency training to mitigate spurious reward when handling lexically similar instructions. Despite it, discriminative RM still requires massive data to address the challenges posed by their inherent sparsity modeling (Sun et al., 2024).

**Generative RM.** GenRM can improve LLMs by synthesizing interpretable critiques. In practice, the LLM-as-a-Judge paradigm (Dubois et al., 2024; Wang et al., 2023; Zheng et al., 2024) demonstrates high consistency between LLMs and humans in evaluation tasks, further highlighting the potential of generative modeling. The self-reward modeling (Yuan et al., 2024b; Wu et al., 2024; Ye et al., 2024) approach showcases the optimization of syn-

thesized critiques through self-improvement. Furthermore, some researchers (Ankner et al., 2024; Yu et al., 2024) try to combine the strengths of both by jointly generating critiques and scalar reward scores. In contrast, as illustrated in Figure 2, our approach emphasizes criteria-based reward modeling by introducing a synthetic preference tree (Yuan et al., 2024a; Lehnert et al., 2024; Tian et al., 2024) to synthesize criteria and incorporating process supervision to reduce error accumulation.

### 3 Preliminaries

Implicit modeling in reward model training reduce data dependency but pose risks like reward hacking<sup>3</sup>. This section explores key challenges (see § 3.1) and suggests potential improvement strategies based on the findings of trials (see § 3.2).

#### 3.1 Barriers to Progress

**Implicit Modeling and Inference.** In reward modeling, the number of required comparisons often far exceeds the available preference pairs (Sun et al., 2024). While sparse comparisons help reduce human annotation costs, they introduce the risk of failing to capture true order relationships, such as non-transitive preferences (Wang et al., 2024a). The human preference ( $y_a$  or  $y_b$ ) can be considered as a weighted combination of multiple  $\hat{\mathcal{R}}$  based on different *criteria*  $\mathcal{C}$ :

$$\begin{aligned}\mathcal{R}(x, y_a, y_b) &= \sum_i^{|C|} \alpha_i \hat{\mathcal{R}}(x, y_a, y_b, c_i), \\ \hat{\mathcal{R}}(x, y_a, y_b, c_i) &\sim \mathcal{P}(y_r | c_i, x, y_a, y_b), \\ c_i &= \mathcal{P}(c | x, y_a, y_b).\end{aligned}\quad (1)$$

However, this weighting coefficient  $\alpha$  is largely determined heuristically, making it impossible to obtain the value. Similarly, BT models and non-CoT GenRMs also rely on the implicit learning of human preferences rather than providing explicit reward signals, making it difficult for them to model complex preference relationships. Considering the overall reward  $\mathcal{R}$  is derived from a combination of criteria-based sub-rewards  $\hat{\mathcal{R}}$ , can we enable the LLMs to generate these sub-rewards, thereby facilitating explicit generative reward modeling?

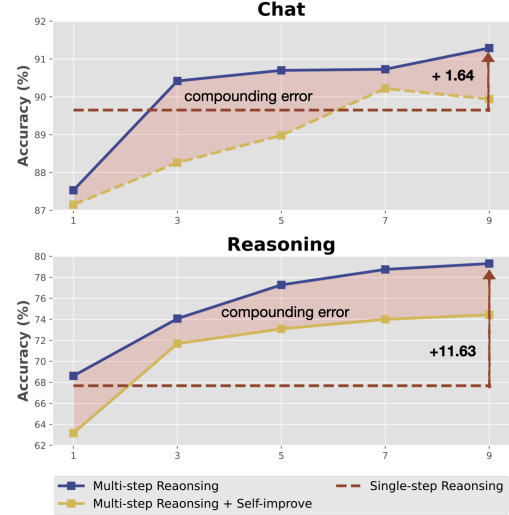


Figure 3: Experimental Results of generative reward models (LLaMA3.1-8B) on RewardBench.

#### 3.2 Trial Experiment Design

To evaluate the effectiveness of criteria-based judgment, we adapt LLaMA3.1-8B-Instruct as a generative reward model. We utilize the Chat and Reasoning datasets from the human preference benchmark RewardBench. First, we guide the model to generate criteria-based CoT judgments using carefully designed prompts and then aggregate all judgments to obtain the final result. As illustrated in Figure 3, single-step reasoning represents the output of a traditional single-step GenRM, while multi-step reasoning corresponds to results derived from 1 to 9 different criteria. Furthermore, we employ rejection sampling to synthesize CoT trajectories on the UltraFeedback dataset for fine-tuning the LLMs, in order to evaluate whether self-improvement (using STaR-like methods (Zelikman et al., 2022)) can further enhance performance.

**Observation 1.** In the experiment, the curve represents the changes in cumulative judgment results based on an increasing number of criteria, demonstrating that multiple criteria-based preference judgment markedly improves model performance. Additionally, incorporating more criteria in the evaluation leads to enhanced accuracy (e.g., achieving an 11.63% improvement in reasoning tasks), surpassing the performance of using a single criterion.

**Observation 2.** Although we aim to optimize the model’s capabilities through the synthesis of CoT

<sup>3</sup>For example, imbalanced label distributions may cause high-frequency labels to dominate predictions, misleading the model’s decisions.

and self-improvement, compounding errors<sup>4</sup> result in a performance decline after fine-tuning.

**Conclusion .** It is evident that aligned LLMs inherently serve as criteria-based reward models, yet this capability still requires further guidance. The primary cause of “compounding errors” is the absence of human supervision in each CoT trajectory. Therefore, our main objective is to improve the quality of criteria-based CoT through explicit training and optimization. Additionally, multi-step reasoning introduces a significant inference cost, which poses challenges for practical deployment.

### 3.3 Motivation

A natural question is that *Can LLMs optimize and prioritize the most critical criteria?* Based on this idea, we reformulated the objective of reward modeling as a further optimization of Equation (1):

$$\begin{aligned} \mathbb{I}\{\mathcal{R}(x, y_a, y_b)\} &= \mathbb{I}\{\hat{\mathcal{R}}(x, y_a, y_b, c^*)\}, \\ c^* &= \arg \max_{c_i} \hat{\mathcal{R}}(x, y_a, y_b, c_i), \end{aligned} \quad (2)$$

where  $c^*$  represents the most critical criterion,  $\mathbb{I}$  represents the indicator function, which takes the value 1 if  $y_a$  is better, and 0 if  $y_b$  is better. This objective requires the explicit optimization of sub-rewards  $\hat{\mathcal{R}}$ , but we lack explicit reward signals in practice. To address this limitation, outcome-based reward learning must be transformed into process-based reward learning. Specifically, in the training data, we only have  $\mathcal{R}$  rather than  $\hat{\mathcal{R}}$ , which must be identified through searching and learning.

**Remark:** *It is important to note that  $c^*$  is not a global optimal solution to the current preference problem, but rather a local optimal solution within the limited criteria candidate space in our work.*

## 4 Methods

To achieve the abovementioned goals, we decompose it into two components: (1) obtaining estimated values for the sub-rewards  $\hat{\mathcal{R}}$ , and (2) optimizing the LLMs to output the most critical criteria  $c^*$ . Thus, we introduce a method **SyncPL**, as shown in Figure 4. This approach first ranks the CoT judgments according to synthesized criteria, generating sub-reward estimates for various trajectories. Then, we fine-tune the LLMs to emphasize critical criteria, ensuring that they are prominently reflected in the model’s predictions.

<sup>4</sup>LLMs experience compounding errors, where initial mistakes in generation propagate as the sequence length increases.

### 4.1 Ranking for Criteria Preference Tree

SyncPL first introduce an additional *rubric* node between the *criterion* node and the *judgment* node, as defined by the following:

$$\begin{aligned} \mathcal{P}(r) &= \mathcal{P}(r \mid x, c_i), \\ \mathcal{P}(y_r) &= \mathcal{P}(y_r \mid x, y_a, y_b, c_i, r_i). \end{aligned} \quad (3)$$

Generative reward models need to generate the sequences of *criteria*, *rubric*, and *judgment* for each trajectory. The *judgment* is a text sequence generated by the model, including explicit evaluation scores ( $\hat{\mathcal{R}}_a, \hat{\mathcal{R}}_b$ ) for the paired responses ( $y_a, y_b$ ). As a result, we can obtain the final preference choice: If  $\hat{\mathcal{R}}_a > \hat{\mathcal{R}}_b$ , then  $y_a$  is the better; if  $\hat{\mathcal{R}}_a < \hat{\mathcal{R}}_b$ , then  $y_b$  is the better. The distributions  $\mathcal{P}$  in Equation 3 are modeled using LLMs.

Given the datasets  $\mathcal{D} = \{(x, y_a, y_b, l)_i\}_{i=1}^N$ , we construct a preference tree using the Monte Carlo rollout policy, where each parent node generates multiple child nodes (two in practice). The preference label  $l$  is annotated by human as ground truth. In the implementation, the sub-tree for each criterion is represented as a complete binary tree, containing 4 distinct evaluation scores. Each path from the root node to a leaf node represents a complete evaluation trajectory  $\mathcal{T} = \{x, y_a, y_b, c_i, r_i, y_r\}$ . It is important to note that we adopt two distinct rollout strategies: For *criteria*, we use LLMs to directly generate text sequence  $\{c_1, \dots, c_{|C|}\}$  relevant to the user query, similar to the brainstorming. For *rubric* and *judgment*, we employ a sampling strategy based on the probabilities of the output.

To optimize the LLMs policy under different criteria, we leverage the oracle label  $l$  and predefined rules to perform credit assignment for various criteria. Unlike traditional Monte Carlo rollout that estimate confidence scores, we do not rely on preference scores for each criterion. Instead, these sub-reward scores are estimated based on the oracle preference and predefined rules:

- **Ranking Rule:** *Criteria* with larger score margins indicate higher confidence in evaluation by LLMs. Consequently, the evaluation trajectory of the current criterion should be generated with higher priority.
- **Consistency Rule:** In each sub-tree with the criterion as the root node, when conflicting judgments arise, the majority voting result can be regarded as final judgments. This suggests that judgments which deviate from the majority are probable to be erroneous.



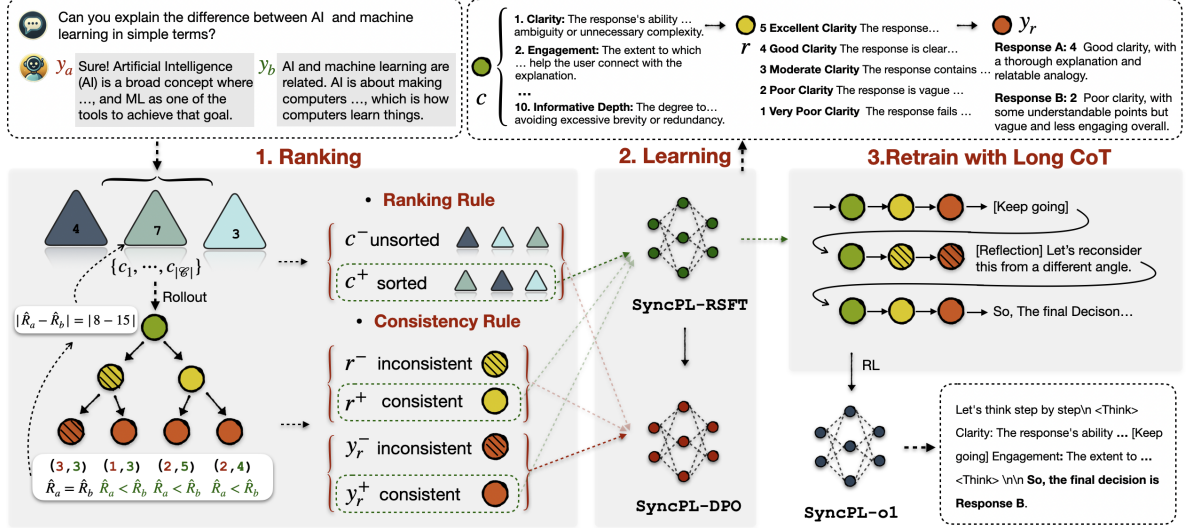


Figure 4: **SyncPL** primarily consists: The **ranking** process focuses on synthesizing preference data, while the **learning** process aims to enhance model through inference-time computations. The **retrain** process can further enforce the reasoning ability by introducing a special prompt: [Keep going] indicates that the previous criteria-based judgment is correct, while [Reflection] signifies an inconsistency between reasoning and the oracle label.

Based on these rules, we can collect pairwise preference training data from different tree node:

**Criteria.** We calculate the score margins ( $|\hat{\mathcal{R}}_a - \hat{\mathcal{R}}_b|$ ) for all criteria and reorder the output sequence  $\{c_1, \dots, c_{|C|}\}$  based on the ranking results ( $|C|$  is set to 10). The reordered sequence is denoted as  $c^+$ , while the original sequence serves as  $c^-$ .

**Rubric & Judgment.** For each *rubric* and *judgment* synthesized based on criterion, we apply the consistency rule to filter out sub-trees: inconsistent results are treated as rejected samples ( $r^-$  and  $y_r^-$ ), while consistent results with large score margins are selected as chosen samples ( $r^+$  and  $y_r^+$ ).

## 4.2 Criteria Preference Learning

In generative reward modeling, high-quality preference pairs ( $c^+, c^-$ ), ( $r^+, r^-$ ), and ( $y_r^+, y_r^-$ ) are utilized for training both RSFT and DPO. Specifically, RSFT employs chosen samples ( $c^+, r^+, y_r^+$ ) as training data, while DPO leverages paired preferences for its training. The reward model can optimize the synthesis process<sup>5</sup> through preference learning, such as the loss function for *criteria* is defined as follows:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, c^+, c^-) \sim \mathcal{D}} \left[ \log \sigma \left( \beta \log \frac{\pi_\theta(c^+|x)}{\pi_{\text{ref}}(c^+|x)} - \beta \log \frac{\pi_\theta(c^-|x)}{\pi_{\text{ref}}(c^-|x)} \right) \right], \quad (4)$$

<sup>5</sup>In practice, we use  $\mathcal{P}_\theta(c|x)$  instead of  $\mathcal{P}_\theta(c|x, y_a, y_b)$  to reduce computational cost.

where  $\pi_{\text{ref}}$  is typically the RSFT model,  $\beta$  is the hyperparameter that controls the proximity of the policy  $\pi_\theta$  to the original policy  $\pi_{\text{ref}}$ .

## 4.3 Retrain with Long CoT Format

OpenAI’s o1-series models introduce inference-time scaling through a long CoT reasoning process. Inspired by this, we also explore utilizing the long CoT for GenRM, leveraging high-quality criteria-based judgments generated by SyncPL-RSFT. Specifically, we reorganize the training data by alternating between chosen and rejected trajectories, aiming to stimulate the model’s reflection ability. Subsequently, we retrain SyncPL-o1 based on LLaMA3.1-8B using DPO as before.

## 4.4 Implementation

Rejection sampling is designed to eliminate low-quality synthetic data by applying a consistency rule to filter out invalid trajectories. This process ensures that only evaluation results consistent with both majority voting and oracle labels are retained. If each sample includes 10 distinct criteria, the number of trajectories required for synthetic data is  $((2 + 4) \times 10)$ . Clearly, this process involves a significant amount of inefficient sampling.

**Efficiency improvement .** To enhance the efficiency of data synthesis, we generate evaluation trajectories for only **one** criterion per dialogue during the RSFT synthesis. For DPO, we synthesize trajectories for all **10** criteria simultaneously to en-

sure comprehensive preference learning. This will result in nearly a  $10\times$  speedup. During DPO, we select only the inconsistent samples between the backbone and RSFT models, resulting in the removal of approximately 70% of the total dataset.

## 5 Experiments

### 5.1 Evaluation Datasets and Details

In experiments, we employ three distinct human preference benchmarks to evaluate the reward models: RewardBench (Lambert et al., 2024), Auto-J (Li et al., 2023), and MT-Bench (Zheng et al., 2024). Specifically, RewardBench comprises four distinct subsets: chat, chat-hard, safety, and reasoning, containing 358, 456, 740, and 1,431 samples, respectively. Each subset is designed to evaluate model performance in helpfulness, instruction-following, safety, and reasoning. Additionally, Auto-J and MT-bench include 1,392 and 3,355 dialogues across diverse domains. We exclude samples labeled as ‘tie’ from the Auto-J. For evaluation, we adopt **consistency** metric between model predictions and human annotations.

For baseline models, we employ the state-of-the-art closed-source generative models GPT-4, GPT-4o, and DeepSeek V3. For the discriminative RM, we select LLaMA3-8B RM with ultrafeedback using the BT objective. For the generative RM, we select Prometheus2, which trained by GPT-4 synthesized data. Additionally, CCloud leverages LLaMA3-405B to generate CoT reasoning paths using a hybrid methods for model training.

### 5.2 Out-of-Distribution Reward Modeling

We utilize Qwen2.5-32B-Instruct-GPTQ-Int8 to construct CoT data and train the LLaMA3.1-8B Distill model as our backbone model. SyncPL leverages dialogue data from UltraFeedback and OffsetBias as training set to construct a fine-grained reward model. It is notable that the data distribution of these training datasets differs significantly from that of the evaluation benchmarks, meaning the results in Table 1 can be regarded as out-of-distribution (OOD) evaluations. With RSFT, SyncPL achieves superior alignment with human preferences, outperforming the LLaMA3.1-8B Distill model by 2.8 points. Moreover, SyncPL-DPO further enhances reward modeling capabilities, yielding a 3.8 point improvement.

SyncPL-o1 is trained on synthetic data generated by SyncPL-SFT, as it can produce rejected data to

better guide the model’s reflection ability compared to SyncPL-DPO. Through reinforcement learning, SyncPL-o1 achieves superior performance, surpassing LLaMA3.1-70B (84.2 v.s. 84.0). While it obtains optimal performance with a single inference step, this primarily depends on SyncPL’s ability to optimize process-level data (criteria-based judgment). The introduction of process supervision signals ensures the correctness of the reasoning chain in long CoT, fully unleashing its potential.

### 5.3 In-Distribution Reward Modeling

To evaluate in-distribution judge performance, we fine-tune Mistral-7B-Instruct using the Fennec (Liang et al., 2024) training data, which includes 3K dialogue samples from Auto-J (Li et al., 2023), along with CoT trajectory data annotated by GPT-4. For a fair comparison, we adopt Agreement (AGR) and Consistency (CNS) as performance metrics, following Auto-J. Notably, CNS measures only positional consistency without considering correctness. As observed in Table 2, the RSFT and DPO models demonstrate significant performance improvements over most baseline methods on both Auto-J and Fennec, achieving the highest agreement score of 58.41.

**Dialogue Correction.** The core capability of GenRM is to identify and rectify flaws in dialogues, thereby enhancing overall quality. To evaluate this, we test our model’s ability to assess and refine dialogues generated by Alpaca-13B (Taori et al., 2023) and LLaMA2-7B Chat (Touvron et al., 2023) in MT-Bench. Unlike previous pairwise evaluations, MT-Bench is a multi-turn dialogue evaluation benchmark that utilizes GPT-4 to score AI responses on a scale from 1 to 10, subsequently establishing ranking relationships based on these scores. Specifically, we generate multiple judgments for the responses and refine those with scores below 3. As shown in Table 3, after refinement, both Alpaca-13B and LLaMA2-7B Chat achieve higher scores. Moreover, the DPO model’s corrections outperform those of the RSFT model, demonstrating that improved evaluation feedback can lead to significant enhancements in evaluation quality.

## 6 Analysis

### 6.1 Scalability of SyncPL’s Branching

To evaluate whether different criteria contribute to performance scalability, we compare model performance across different training stages using














Evaluator	Backbone	Reward Bench					Auto-J	MT-Bench
		Chat	Chat H.	Safety	Reasoning	Avg.		
GPT4-o		96.1	76.1	86.6	88.1	86.7	74.6	80.7
GPT4		95.3	74.3	87.6	86.9	86.0	73.1	79.9
DeepSeek-V3		96.9	79.8	87.0	84.3	87.0	-	-
Prometheus2-7B		85.5	49.1	78.7	76.5	72.4	75.1	74.6
Prometheus2-8x7B		93.0	47.1	83.5	77.4	75.3	80.0	73.6
LLaMA3-8B RM		94.7	44.9	66.2	83.6	72.4	79.3	78.1
C4o		94.3	46.5	78.6	84.6	76.0	<b>81.2</b>	<b>79.9</b>
LLaMA3.1-8B		80.7	49.8	64.0	68.1	65.7	-	-
LLaMA3.1-70B		<b>97.2</b>	70.2	82.8	<b>86.0</b>	84.0	-	-
LLaMA3.1-8B Distill†		92.9	61.8	82.5	79.9	79.3	76.5	73.2
<b>SyncPL-RSFT-8B</b>		94.1 <sub>0.16</sub>	66.4 <sub>0.25</sub>	84.9 <sub>0.20</sub>	83.0 <sub>0.73</sub>	82.1	78.0	74.8
<b>SyncPL-DPO-8B</b>		93.3 <sub>0.15</sub>	71.3 <sub>0.33</sub>	<b>86.5</b> <sub>0.15</sub>	81.2 <sub>0.53</sub>	83.1	77.1	74.1
<b>SyncPL-o1-8B</b>		93.9 <sub>0.19</sub>	<b>73.2</b> <sub>0.93</sub>	85.8 <sub>0.77</sub>	83.7 <sub>1.12</sub>	<b>84.2</b>	77.8	73.5

Table 1: The experimental results on RewardBench, Auto-J, and MT-Bench, where **bold numbers** indicate the best performance among open-source, small-size LLMs. † represent our reimplementations.

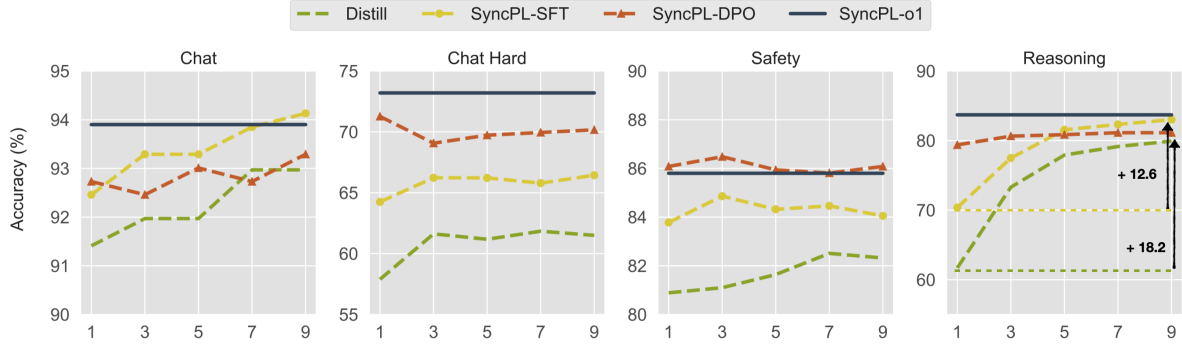


Figure 5: The results of judgments ensemble based on different criteria. **SyncPL-o1** utilizes long CoT for preference judgments and inherently incorporates multiple criteria. One key advantage of an o1-like model is that it requires only a single inference step, whereas other models must perform multiple steps depending on the number of criteria.

Evaluator	C	w/ Tie		w/o Tie	
		AGR ↑	CNS ↑	AGR ↑	CNS ↑
GPT-4	1	62.28	86.28	-	-
GPT-3.5	1	44.41	72.39	59.86	73.57
Auto-J †	1	55.13	82.44	74.13	87.26
Fennec	1	55.36	83.80	68.63	86.33
	5	55.80	85.52	74.14	89.19
<b>SyncPL-RSFT</b>	1	56.87	87.92	70.92	88.99
	5	56.32	87.86	75.17	89.79
<b>SyncPL-DPO</b>	1	57.54	90.66	78.60	93.32
	5	<b>58.41</b>	<b>91.52</b>	<b>79.69</b>	<b>93.72</b>

Table 2: For a fair comparison on the Auto-J benchmark, we use only five criteria (C). Additionally, methods marked with † represent our reimplementations.

RewardBench. As shown in Figure 5, both the Distill model and SyncPL-SFT exhibit a significant performance improvement as the number of criteria increases. This effect is particularly notable in rea-

Models	Rating	Refine Rate
GPT-4	<b>8.96</b>	-
LLaMA2-70B Chat	6.99	-
LLaMA2-7B Chat	6.26	-
w/ <b>SyncPL-RSFT</b> Corr	6.85	87.5%
w/ <b>SyncPL-DPO</b> Corr	<b>7.08</b>	72.5%
Alpaca-13B	4.97	-
w/ <b>SyncPL-RSFT</b> Corr	6.61	95.0%
w/ <b>SyncPL-DPO</b> Corr	<b>6.85</b>	<b>98.8%</b>

Table 3: Results of dialogue correction.

soning tasks, where different criteria significantly enhance the model’s judgment accuracy, with maximum improvements of +18.2% and +12.6%, respectively. Furthermore, the SyncPL-DPO model optimizes the ability to select key criteria more effectively, allowing it to make accurate judgments with only single criterion. This demonstrates that our preference optimization method effectively substantiates the claims presented in the motivation.

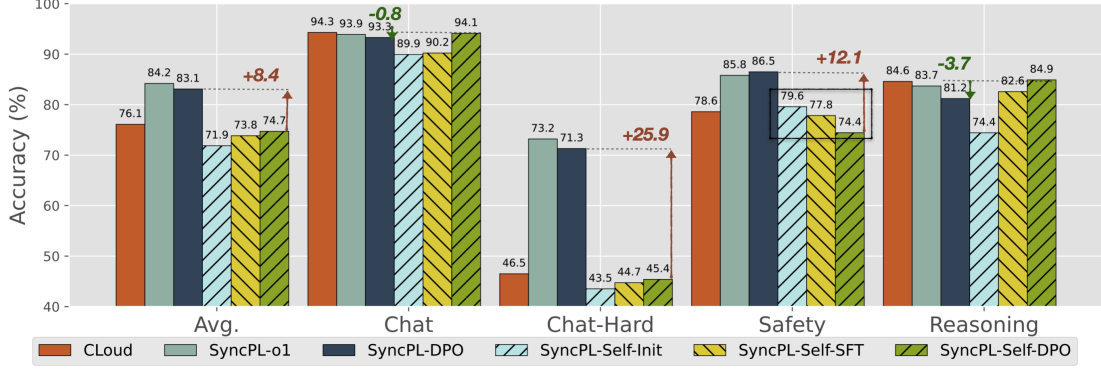


Figure 6: The results demonstrate the effectiveness of self-synthesized CoT data in combination with SyncPL.

Models	Chat	Chat H.	Safety	Reasoning	Avg.
SyncPL-RSFT	94.1	66.4	84.9	83.0	<b>82.1</b>
w/ UltraF.	<b>94.7</b>	52.2	81.0	<b>83.4</b>	77.9
w/ OffsetB.	89.9	<b>67.7</b>	<b>85.4</b>	82.1	80.7
SyncPL-DPO	<b>93.3</b> <sub>↓0.8</sub>	<b>71.3</b> <sub>↑4.9</sub>	<b>86.5</b> <sub>↑1.6</sub>	81.2 <sub>↓1.8</sub>	<b>83.1</b>
w/ UltraF.	94.9 <sub>↑0.2</sub>	54.4 <sub>↑2.2</sub>	83.1 <sub>↑2.1</sub>	<b>82.6</b> <sub>↓0.8</sub>	78.8
w/ OffsetB.	85.5 <sub>↓4.4</sub>	71.2 <sub>↑3.5</sub>	85.3 <sub>↓0.1</sub>	77.1 <sub>↓5.0</sub>	79.8

Table 4: Results on different dialogue datasets (Offset-Bias and UltraFeedback) based on LLaMA3.1-8B.

Additionally, the SyncPL-o1 model employs a long CoT format, leveraging the key criteria extracted by SyncPL-SFT. This method proves to be more optimization-friendly in experiments while ensuring stable performance improvements. It can be observed that SyncPL-o1 consistently achieves superior results compared to others across almost all sub-datasets.

## 6.2 Ablations of Different Dialogue Datasets

As shown in Figure 4, we find that OffsetBias significantly improves evaluation performance on Chat Hard but does not outperform UltraFeedback in Chat evaluations. The primary reason for this is the **distribution shift** between Chat Hard and other datasets. Using OffsetBias alone as the training set leads to reward hacking, causing performance degradation across all datasets except Chat Hard. To address this issue, we combined UltraFeedback and OffsetBias for training in our previous experiments. Nevertheless, the distribution gap affects the DPO training process, emphasizing the need for more suitable dialogue datasets in future research.

## 6.3 SyncPL with Self-Improvement

In this subsection, we further examine the necessity of leveraging data distillation from larger models to warmup the backbone model. To investigate this, we retrain the SyncPL-self series models

using only self-synthesized CoT data generated by LLaMA3.1-8B. As shown in Figure 6, the results reveal that small size LLMs struggle to enhance certain skills through test-time scaling, as also observed by Guo et al. (2025). Moreover, our findings indicate a substantial performance gap between SyncPL-self and the distilled model, particularly in Chat and Safety tasks, with a degradation of up to **25.9** points. A major factor is the presence of bias in Chat Hard, which makes small size models prone to misjudgment. These results highlight the critical importance of high-quality CoT data in improving the reasoning of small size LLMs.

Additionally, we observe that even with self-improvement, SyncPL-self achieves consistent performance gains on Rewardbench (except for safety, likely due to “alignment tax”). Notably, it surpasses SyncPL-o1 in chat and reasoning sub-tasks.

## 6.4 Error Analysis

We conduct an analysis of error types in chat datasets, revealing that **60%** of errors stem from the preference for shorter responses, 13% are attributed to intent misinterpretation and inherent model limitations, and 27% involve high ambiguity. For more details, please refer to the Appendix A.2.

## 7 Conclusion

In this work, we introduce SyncPL, which reformulates the generative reward modeling process into a criteria-based preference tree, enabling LLMs to more effectively identify key criteria. Our experiments not only validate the efficacy of multi-criteria training but also demonstrate that SyncPL-o1 achieves the most substantial gains in the long-CoT format, outperforming even the LLaMA3.1-70B model. Beyond improving performance, our approach introduces a effective mechanism for process supervision in generative modeling, offering a



promising solution to mitigate compounding errors stemming from model limitations.

## 8 Limitations

**The Candidate Space of Criteria .** A intuitive limitation lies in the candidate space of criteria that LLMs consider during judgments, which directly affects whether the model can select the optimal criterion for making judgments. In our work, we rely on LLMs to autonomously generate criteria for evaluation, making the effectiveness of this approach partially dependent on the model’s inherent capabilities. Possible ways to expand the candidate space include, but are not limited to: (1) Using more strong LLMs for data synthesis. (2) Constructing a candidate pool list and retrieving relevant criteria as needed. Since this study primarily aims to validate the feasibility of criteria-based preference learning, we do not extensively explore these potential extensions.

**Pure RL v.s. Process Supervision .** The key difference from DeepSeek R-zero lies in the learning approach. R-zero relies solely on reinforcement learning, optimizing through rule-based rewards. In contrast, our method first leverages process supervision to generate effective intermediate trajectories before applying reinforcement learning for further improvement. Notably, the R-zero study demonstrates that their approach is ineffective on small-scale LLMs. In contrast, our method is specifically designed to enhance the performance of these models. Moreover, since reinforcement learning necessitates a rigorously defined candidate set for effective optimization, the training process can quickly collapse if high-quality solutions are rarely sampled. Our method addresses this challenge by filtering candidates based on multiple criteria, thereby ensuring a search domain that is highly conducive to optimization.

## Acknowledgments

We want to thank all the anonymous reviewers for their valuable comments. This work was supported by the National Science Foundation of China (NSFC No. 62206194), the Natural Science Foundation of Jiangsu Province, China (Grant No. BK20220488), the Young Elite Scientists Sponsorship Program by CAST (2023QNRC001), and the Priority Academic Program Development of Jiangsu Higher Education Institutions. We also acknowledge MetaStone Tech. Co. for providing

us with the software, optimisation on high performance computing and computational resources required by this work.

## References

- Zachary Ankner, Mansheej Paul, Brandon Cui, Jonathan D Chang, and Prithviraj Ammanabrolu. 2024. Critique-out-loud reward models. *arXiv preprint arXiv:2408.11791*.
- Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.
- Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.
- Lichang Chen, Chen Zhu, Davit Soselia, Jiuhai Chen, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Odin: Disentangled reward mitigates hacking in rlhf. *arXiv preprint arXiv:2402.07319*.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.
- Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. 2023. Reward model ensembles help mitigate overoptimization. *arXiv preprint arXiv:2310.02743*.
- Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.
- Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2024. AlpacaFarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

- Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. 2023. Prometheus: Inducing fine-grained evaluation capability in language models. *arXiv preprint arXiv:2310.08491*.
- Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. 2024. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*.
- Lucas Lehnert, Sainbayar Sukhbaatar, Paul Mcvay, Michael Rabbat, and Yuandong Tian. 2024. Beyond a\*: Better planning with transformers via search dynamics bootstrapping. *arXiv preprint arXiv:2402.14083*.
- Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. 2023. Generative judge for evaluating alignment. *arXiv preprint arXiv:2310.05470*.
- Xiaobo Liang, Haoke Zhang, Helan hu, Juntao Li, Jun Xu, and Min Zhang. 2024. [Fennec: Fine-grained language model evaluation and correction extended through branching and bridging](#). *Preprint*, arXiv:2405.12163.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522.
- R Duncan Luce. 1959. *Individual choice behavior*, volume 4. Wiley New York.
- 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, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Alizée Pace, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2024. West-of-n: Synthetic preference generation for improved reward modeling. *arXiv preprint arXiv:2401.12086*.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.
- Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.
- Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.
- Swarnadeep Saha, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. 2023. Branch-solve-merge improves large language model evaluation and generation. *arXiv preprint arXiv:2310.15123*.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Jiaming Shen, Ran Xu, Yennie Jun, Zhen Qin, Tianqi Liu, Carl Yang, Yi Liang, Simon Baumgartner, and Michael Bendersky. 2024. Boosting reward model with preference-conditional multi-aspect synthetic data generation. *arXiv preprint arXiv:2407.16008*.
- Lingfeng Shen, Sihao Chen, Linfeng Song, Lifeng Jin, Baolin Peng, Haitao Mi, Daniel Khoshnabi, and Dong Yu. 2023. The trickle-down impact of reward (in-) consistency on rlhf. *arXiv preprint arXiv:2309.16155*.
- Roger N Shepard. 1957. Stimulus and response generalization: A stochastic model relating generalization to distance in psychological space. *Psychometrika*, 22(4):325–345.
- Joar Skalse, Nikolaus Howe, Dmitrii Krashennnikov, and David Krueger. 2022. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471.
- Hao Sun, Yunyi Shen, and Jean-Francois Ton. 2024. Rethinking bradley-terry models in preference-based reward modeling: Foundations, theory, and alternatives. *arXiv preprint arXiv:2411.04991*.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. [https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca).
- Ye Tian, Baolin Peng, Linfeng Song, Lifeng Jin, Dian Yu, Haitao Mi, and Dong Yu. 2024. Toward self-improvement of llms via imagination, searching, and criticizing. *arXiv preprint arXiv:2404.12253*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti

- Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Mingzhi Wang, Chengdong Ma, Qizhi Chen, Linjian Meng, Yang Han, Jiancong Xiao, Zhaowei Zhang, Jing Huo, Weijie J Su, and Yaodong Yang. 2024a. Magnetic preference optimization: Achieving last-iterate convergence for language models alignment. *arXiv preprint arXiv:2410.16714*.
- Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, et al. 2023. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. *arXiv preprint arXiv:2306.05087*.
- Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024b. Helpsteer2: Open-source dataset for training top-performing reward models. *arXiv preprint arXiv:2406.08673*.
- Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar. 2024. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. *arXiv preprint arXiv:2407.19594*.
- Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. 2024. Regularizing hidden states enables learning generalizable reward model for llms. *arXiv preprint arXiv:2406.10216*.
- Zihuiwen Ye, Fraser Greenlee-Scott, Max Bartolo, Phil Blunsom, Jon Ander Campos, and Matthias Gall  . 2024. Improving reward models with synthetic critiques. *arXiv preprint arXiv:2405.20850*.
- Yue Yu, Zhengxing Chen, Aston Zhang, Liang Tan, Chenguang Zhu, Richard Yuanzhe Pang, Yundi Qian, Xuewei Wang, Suchin Gururangan, Chao Zhang, et al. 2024. Self-generated critiques boost reward modeling for language models. *arXiv preprint arXiv:2411.16646*.
- Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, et al. 2024a. Advancing llm reasoning generalists with preference trees. *arXiv preprint arXiv:2404.02078*.
- Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024b. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.
- 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*.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

## A Appendix

### A.1 Training details

We utilize the DeepSpeed (Rasley et al., 2020) library, Zero Redundancy Optimizer (ZeRO) (Rajbhandari et al., 2020) Stage 3, and FlashAttention (Dao, 2023), along with a mixed precision computation approach using bfloat16 (BF16) and tfloat32 (TF32), across 8 NVIDIA A100 GPUs. For optimization, we set the number of gradient accumulation steps to 32. The initial and SFT models are trained with a learning rate of  $5e-6$ , while the DPO model is trained with a learning rate of  $5e-7$ . Each training stage consists of one epoch, and we set  $\beta = 0.1$  for training the DPO model. The prompt templates used in our evaluation are provided in Table 6.

### A.2 Case Study

#### Case 1: Length Bias

**Query:** Who created the Superman cartoon character?

**Response A:** The Superman cartoon character was created by Jerry Siegel and Joe Shuster in 1938.

**Response B : (ground truth)** Superman, the iconic comic book superhero, was created by writer Jerry Siegel and artist Joe Shuster. Superman first appeared in Action Comics 1, which was published by Detective Comics, Inc. (later DC Comics) in June 1938. The character's immense popularity established him as one of the most enduring and recognizable figures in the superhero genre.

**Judge :** Accuracy of Information: This criterion ensures that the response correctly identifies the creators of the Superman ...

**Response A** directly answers the user query by naming Jerry Siegel and Joe Shuster as the creators of the Superman cartoon character. ... The response is clear and to the point, making it easy for the user to understand the answer. ... **Response B** also correctly identifies Jerry Siegel and Joe Shuster as the creators of Superman. However, it provides additional information about the character's first appearance and the publisher, which, while interesting, is not directly asked for in the user query. ....

#### Case 2: Length Bias

**Query:** what language does argentina people speak?

**Response A:** The official language of Argentina is Spanish.

**Response B : (ground truth)** The primary language spoken in Argentina is Spanish. Specifically, the variety of Spanish spoken there is known as Rioplatense Spanish, which is characterized by its unique accent and the use of the pronoun "vos" instead of "tú" for the second person singular. Additionally, there are various indigenous languages spoken in different regions of the country, as well as immigrant languages such as Italian, German, and English, but these are much less widespread than Spanish.

**Judge :** Accuracy of Information: This criterion ensures that the response correctly identifies the primary language spoken in Argentina. It is crucial because providing incorrect information can lead to misinformation and dissatisfaction with the response. ...

**Response A** directly answers the user query by stating that the official language of Argentina is Spanish. This response is concise and to the point, which aligns well with the user's query. It does not provide additional information that might not be relevant to the user's query, which can sometimes be seen as a positive aspect of brevity. ... **Response B** provides more detailed information about the language spoken in Argentina, including the specific variety of Spanish and the use of indigenous and immigrant languages. While this information might be interesting and informative, it goes beyond the scope of the user's query, which was simply to identify the primary language spoken in Argentina. This makes the response less focused and potentially less useful for the user's immediate needs. Comparison: Response A is more aligned with the user's query, providing a direct and concise answer to the question. Response B, while informative, includes additional details that were not requested by the user, which can be seen as unnecessary and potentially confusing. ...



### Case 3: Length Bias

**Query:** You are given a paper citation, convert it to the requested citation style. Chicago: Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N., Kaiser, Lukasz, and Illia Polosukhin. "Attention Is All You Need." arXiv, (2017). <https://doi.org/10.48550/arXiv.1706.03762>. MLA:

**Response A:** Vaswani, Ashish, et al. "Attention Is All You Need." ArXiv, 17 June 2017, <https://doi.org/10.48550/arXiv.1706.03762>.

**Response B :** (ground truth) Vaswani, Ashish, et al. "Attention Is All You Need." arXiv, 2017, doi:10.48550/arXiv.1706.03762. In the MLA citation style, the citation would be formatted as follows: Author(s) of the source: Vaswani, Ashish, et al. Title of the source: "Attention Is All You Need." Title of the container (if the source is part of a larger work): arXiv Publication date: 2017 Location (such as a URL or page range): doi:10.48550/arXiv.1706.03762 Note that in MLA style, the authors are listed with the last name followed by the first name, and the names are separated by a comma. ...

**Judge :** Accuracy of Citation Style: This criterion evaluates whether the response correctly converts the citation to the MLA style as requested. ... **Response A provides a direct conversion of the Chicago citation style to MLA style, adhering closely to the MLA format guidelines. It includes the author's last name, first name, title of the source, title of the container, publication date, and DOI. The response is concise and directly answers the user query without unnecessary elaboration. ... Response B attempts to explain the MLA citation style but does not provide a direct conversion of the citation as requested. Instead, it offers a general explanation of how to format a citation in MLA style, which is not what the user query asked for. The response is informative but does not fulfill the specific request for a citation in MLA style. ...**

### Case 4: Model Capability Limitation

**Query:** You should capitalize the sentence according to the guide. Guide: Every other letter alternates between lower case and upper case. Sentence: A giant spider blocks your path.

**Response A: (ground truth)** Sure! Here's the sentence with alternating capitalization: A gIaNt SpIdEr BlOcKs YoUr PaTh.

**Response B :** A GiANT spider BLOCKS your path.

**Judge :** Consistency with the Guide: The response must adhere strictly to the guide provided, ensuring that every other letter is alternately in lower case and upper case. This criterion is crucial as it directly measures the AI's ability to follow instructions accurately. Judge: Response A Score: 1 Analysis of Response A: **Response A does not follow the guide provided in the user query. The guide specifies that every other letter should alternate between lower case and upper case, but Response A does not maintain this pattern. Instead, it seems to randomly capitalize letters, which does not meet the criteria set by the user.** Response B Score: 5 Analysis of Response B: **Response B correctly follows the guide provided in the user query. It alternates between lower case and upper case letters as specified, maintaining the pattern throughout the sentence. This response accurately capitalizes the sentence according to the guide, making it the superior response.** Comparison: Response A fails to adhere to the guide's instructions, leading to a score of 1. In contrast, Response B successfully alternates between lower case and upper case letters as instructed, earning a score of 5. Response B is superior as it accurately follows the guide and provides the correct capitalization of the sentence. ... Relevance to the Query: ... Accuracy of Capitalization: ... Response A does not maintain this pattern. Instead, it randomly capitalizes letters, which does not meet the criteria set by the user. ... It alternates between lower case and upper case letters as specified, maintaining the pattern throughout the sentence. ...

So, the final decision is Response B

### Case 5: High Ambiguity

**Query:** I want you to act as a Tiktok Ads Keywords generator. Your task is to come up with keywords that can be used in Tiktok advertising campaigns. The keywords should be relevant to the target audience and the product or service being advertised. Your responses should not include more than 5 keywords per request.

**Response A: (ground truth)** Sure, I'd be happy to help! Can you please provide me with some information about the product or service being advertised, as well as the target audience? This will help me generate keywords that are relevant and effective for your TikTok advertising campaign.

**Response B :** TikTok is a popular short-form video sharing app that ... Here are a few TikTok ads keywords generator ideas: 1. Trends and Hashtags: TikTok users are always on the lookout for the latest trends and hashtags. By incorporating these into your ads, you can increase the chances of your ads being seen by the right audience. ... 5. Comedy: TikTok users love to laugh, so if your ad is funny, you can use comedy-related keywords to target TikTok users. 6. Gaming: TikTok users also love gaming, so if your ad is about a new game, you can use gaming-related keywords to target. ...

**Judge :** Relevance to the Query: ... **Response A** is a good start as it acknowledges the need for more information about the product or service and the target audience. However, it does not provide any specific keywords or a list of ideas as requested in the user query. It is more of a request for additional information rather than a direct response to the query. ... **Response B** provides a comprehensive overview of the types of keywords that can be used in TikTok advertising campaigns. It includes a variety of categories such as trends, music, dancing, fashion, comedy, gaming, celebrities, animals, contests, and quotes. This response is more aligned with the user query's request for a list of keywords and ideas. However, it could have been more concise and directly provided the requested 5 keywords per request as specified in the user query. ...

As illustrated above, we identify several types of errors observed during inference. The majority of these errors (from Case 1 to Case 3) arise from a preference for shorter responses. However, these shorter responses are not necessarily incorrect but may simply reflect a misalignment with human preferences at the time, which favored longer answers. Additionally, the error in Case 4 stems from the model's inability to distinguish variations in letter casing, leading to misjudgments in specific cases. Finally, we present a scenario where both responses have flaws in Case 5. However, since a choice must be made, the ground truth is set to A. In fact, Response B also provides a concrete answer and could be considered more helpful than A in certain contexts. This issue primarily stems from the ambiguity in criteria selection, as different judges may prioritize different key criteria, leading to variations in judgment.

Evaluator	Reward Bench				
	Chat	Chat H.	Safety	Reasoning	Avg.
GPT3.5-turbo	92.2	44.5	65.5	59.1	65.3
LLaMA3-8B	85.5	41.6	68.0	64.8	65.0
SyncPL-o1-16B	83.8	49.8	67.4	67.9	67.3

Table 5: The experimental results on RewardBench.

### A.3 Results of MoE models

We train an MoE model and evaluate it on RewardBench based on DeepSeek-V2-Lite, which consists of 16B total parameters with 2.4B active parameters. As shown in Table 5, despite relying on only 2.4B active parameters, it still outperforms strong models such as LLaMA3-8B, demonstrating the generalizability of our approach.

Step	Content
Criteria	<p>For evaluating human satisfaction with responses from an AI assistant based on a [User Query], we need to brainstorm and establish ten [Evaluation Criteria] directly linked to the user's query. These criteria play a crucial role in objectively assessing response content, with higher priority and greater evaluation weight.</p> <p>***</p> <p>As an illustration:</p> <ol style="list-style-type: none"> <li>1. Relevance: Evaluate whether the response is directly related to the user's query.</li> <li>2. Criterion: Assess the correctness of the information provided in the response. etc.</li> </ol> <p>***</p> <p>[User Query]: {query}</p> <p>***</p> <p>Please return ten [Evaluation Criteria]:</p>
Rubrics	<p>Consider a [User Query] and [Evaluation Criteria] for evaluating response satisfaction. Reflect on these criteria and offer a comprehensive [Scoring Guideline] on a scale of 1-5 (1 represents 'Not at all satisfactory' and 5 represents 'Extremely satisfactory'). Ensure that these guidelines are closely tied to both the user query and the assessment criteria, allowing for a precise evaluation of possible responses to the user query. Conduct a detailed comparison of the [Scoring Guideline] to ease adherence and assist individuals in assigning reasonable scores.</p> <p>***</p> <p>[User Query]: {query}</p> <p>***</p> <p>[Evaluation Criteria]: {criteria}</p> <p>***</p> <p>Please return detailed [Scoring Guideline]:</p>
Judgments	<p>Given a [Dialogue Context] and a [User Query], please score the responses (A and B) from two AI assistants according to the [Evaluation Criteria] and [Scoring Guideline]. Ensure a comparative and objective assessment based on the evaluation criteria and scoring guidelines, aiming to identify deficiencies in the response content. Provide a final score of 1-5 along with relevant explanations.</p> <p>***</p> <p>[Dialogue Context]: {context}</p> <p>***</p> <p>[User Query]: {query}</p> <p>***</p> <p>[Evaluation Criteria]: {criteria}</p> <p>***</p> <p>[Scoring Guideline]: {scoring guidelines}</p> <p>***</p> <p>[The Start of Response A]:{response1}[The End of Response A]</p> <p>***</p> <p>[The Start of Response B]:{response2}[The End of Response B]</p> <p>***</p> <p>Please return [Judge Result] as follows:  Response A Score: 3 Analysis of Response A: Explanation of the score for the Response A.  Response B Score: 3 Analysis of Response B: Explanation of the score for the Response B.  Comparison: Discuss the comparative strengths and weaknesses of Response A and Response B [Judge Result]:</p>

Table 6: Prompts for multi-step evaluation. The criteria in Scoring Guidelines and Pairwise-eval is regularly extracted from the output of the first step. The scoring guidelines in Pairwise-eval are the output of the second step.