APLOT: Robust Reward Modeling via Adaptive Preference Learning with Optimal Transport

Zhuo Li[1,2,3], **Yuege Feng**[4], **Dandan Guo**[5,6]*, **Jinpeng Hu**[7], **Anningzhe Gao**[1]†, **Xiang Wan**[1],
[1] Shenzhen International Center for Industrial and Applied Mathematics,
[2] Shenzhen Research Institute of Big Data,
[3] The Chinese University of Hong Kong, Shenzhen, [4] Birmingham City University,
[5] Jilin University, [6] KAUST, [7] Hefei University of Technology,

Abstract

The reward model (RM) plays a crucial role in aligning Large Language Models (LLMs) with human preferences through Reinforcement Learning, where the Bradley-Terry (BT) objective has been recognized as simple yet powerful, specifically for pairwise preference learning. However, BT-based RMs often struggle to effectively distinguish between similar preference responses, leading to insufficient separation between preferred and non-preferred outputs. Consequently, they may easily overfit easy samples and cannot generalize well to Out-Of-Distribution (OOD) samples, resulting in suboptimal performance. To address these challenges, this paper introduces an effective enhancement to BT-based RMs through an adaptive margin mechanism. Specifically, we design to dynamically adjust the RM focus on more challenging samples through margins, based on both semantic similarity and model-predicted reward differences, which is approached from a distributional perspective solvable with Optimal Transport (OT). By incorporating these factors into a principled OT cost matrix design, our adaptive margin enables the RM to better capture distributional differences between chosen and rejected responses, yielding significant improvements in performance, convergence speed, and generalization capabilities. Experimental results across multiple benchmarks demonstrate that our method outperforms several existing RM techniques, showcasing enhanced performance in both In-Distribution (ID) and OOD settings. Moreover, RLHF experiments support our practical effectiveness in better aligning LLMs with human preferences. Our code is available at https://github.com/BIRlz/APLOT.

1 Introduction

Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022; Rafailov et al., 2024; DeepSeek-AI et al., 2025) has emerged as a particularly effective approach in improving the effectiveness and helpfulness of Large Language Models (LLMs) (OpenAI, 2024; Touvron et al., 2023; Yang et al., 2024a), and achieving better alignment with human preferences in various fields of artificial intelligence (AI) (OpenAI, 2023; Cobbe et al., 2021b; Shao et al., 2024; Suzgun et al., 2022a; Hu et al., 2025b,a; Dai et al., 2025; Hu et al., 2023, 2022; Li et al., 2025b; Hu et al., 2024). RLHF begins with optimizing a reward model (RM), which produces feedback that quantifies the quality and correctness of users' preferences of the provided responses, and thus maximizing reward will direct the LLMs to model effectively satisfy human queries (Ouyang et al., 2022).

Current RM methods can be broadly categorized into discriminative (Ouyang et al., 2022) and generative approaches (Zheng et al., 2023). Among discriminative methods, a prevalent strategy involves pairwise comparison-based learning, which aims to rank preferred and non-preferred responses based on human annotation by leveraging implicit objectives, such as the Bradley-Terry (BT) model (Bradley and Terry, 1952). While BT model has achieved certain successes, it still faces several limitations, including "over-optimization" (Gao et al., 2023b; Coste et al., 2023) that describes a phenomenon where the policy optimization strategy seemingly enhances the proxy reward model but actually leads to the degeneration of the true reward function.

To address this, several studies have focused on enhancing the reward model with constrained proxy optimization (Dubois et al., 2023; Yang et al., 2024b; Chan et al., 2024; Touvron et al., 2023) or ensemble techniques (Yang et al., 2024c; Wang et al., 2024b; Coste et al., 2023; Eisenstein et al., 2023). However, these resulting reward models still struggle to distinguish between similar responses, especially when the reward differences are subtle,

*Co-corresponding author, *guodandan@jlu.edu.cn*.
†Co-corresponding author, *anningzhegao@gmail.com*.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 5525–5539
November 4-9, 2025

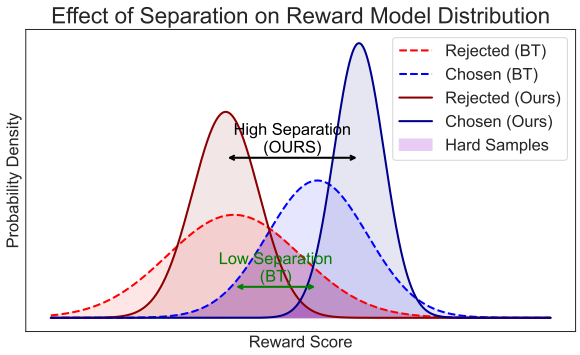

Figure 1: Illustration of the limitation of the traditional BT-based reward model, which only enforces higher scores for chosen samples over rejected ones, neglecting the magnitude of the score difference and resulting in low separation between reward distributions, particularly for hard samples. Our method achieves significantly improved separation, leading to better reward modeling.

leading to insufficient separation between preferred and non-preferred responses, resulting in suboptimal model performance and over-fitting to easy samples (Yang et al., 2024b; Wang et al., 2024a). As shown in Figure 1, BT-based preference learning methods stem from the idea of ranking and only require that chosen samples receive higher scores than rejected samples. This approach neglects the relative magnitude of the score difference, leading to the observed low separation between reward distributions, especially for hard samples.

In this paper, we introduce an adaptive margin to enhance the pairwise BT reward model that is formulated from a distribution-aware perspective using Optimal Transport (OT) (Cuturi, 2013), enabling improved differentiation between preference responses. Our core idea is to dynamically adjust the learning difficult of each training triplet through adaptive margin based on its semantic similarity and model-predicted reward difference. As a result, our proposed method yields a significantly higher separation, as shown in Figure 1, and enables more effective discrimination between positive and negative examples for improved reward modeling, ensuring that the RM focuses more on challenging samples while avoiding overfitting on easier ones.

Specifically, we model the margin between the distribution of chosen responses and that of rejected responses as an OT distance, which naturally captures the distributional differences between the two response types. By incorporating both semantic similarity and reward differences into the cost matrix design, OT provides a principled way to estimate desired margins in an adaptive way. Finally, we can incorporate margins into preference learning objective to optimize an improved RM with better performance and robustness in both In-Distribution (ID) and Out-Of-Distribution (OOD) settings. Additional, our approach outperforms several popular RMs across multiple benchmarks, validating its effectiveness and practical utility. Moreover, we observe that our method helps with faster convergence speed without significant additional training consumption. We summarize our contributions as follows:

1. We propose a novel adaptive margin mechanism to improve pairwise reward models, formulated from a distribution-aware perspective using OT.

2. Our approach enhances the reward model's ability to better distinguish between similar preference responses by adaptively focusing on challenging samples, by considering both semantic similarity and predicted reward difference.

3. Experiments show that our method achieves significant improved performance and robustness in both ID and OOD settings on multiple benchmarks, along with faster convergence speed, validating its effectiveness and practical utility.

2 Related Work

Reward modeling is a critical component in preference learning and RLHF, broadly categorized into discriminative and generative approaches. For the former, classical methods can be traced back to the Bradley-Terry model (Bradley and Terry, 1952) and Plackett-Luce model (Plackett, 1975; Luce, 1959), which optimize an implicit reward function (e.g., a classifier) by learning to imitate human preferences in a pairwise or listwise ranking loss, respectively. Recent studies are more centralized around designing advanced reward models by introducing multi-objective reward functions (Yang et al., 2024c; Wang et al., 2024b), increasing the quality and quantity of training samples (Dubois et al., 2023), regularizing hidden states (Yang et al., 2024b), learning dense rewards (Chan et al., 2024), and causal learning (Liu et al., 2024b). In addition, several popular preference datasets are proposed to help train a robust reward model, such as the Unified-Feedback (UF) Preference dataset[1] and the

[1] https://huggingface.co/datasets/llm-blender/Unified-Feedback

Skywork Preference dataset-80K (SP) (Liu et al., 2024a).

Generative reward modeling is an alternative to classifier-based discriminative reward models by directly employing an LLM to generate a judgment between responses. Generative models excel in providing nuanced, interpretable assessments, capturing subtle differences in language use, and offering deeper insights into the decision-making process (Liu et al., 2024a). In addition, some works have emerged to fine-tune models specifically for the task of rating or choosing responses from LLMs (Wang et al., 2024c) and others use the policy LM itself as a generative reward model via prompting it to behave as a judge, in order to achieve better performance of generative reward models (Wang et al., 2024d). Our work is more in line with the discriminative approach.

Adaptive Margin Estimation has been explored in various machine learning domains. In metric learning, adaptive margins are used to enforce varying separation distances between classes based on their intrinsic similarity (Sohn, 2016; Wu et al., 2017). For instance, Sohn (2016) proposed a dynamic margin for triplet loss, where the margin is adjusted based on the difficulty of the triplet. Similarly, in contrastive learning, adaptive margins have been used to improve the discriminative power of learned representations (Khosla et al., 2020). In the context of preference learning, adaptive margins have been less explored but hold significant potential. Touvron et al. (2023) introduces a margin-based regularization term in vanilla BT training objective to help differentiate preferred and non-preferred responses. However, this approach relies on fixed or heuristic margin assignments by requiring additional human annotations, which introduces more consumption and may not fully capture the nuanced differences between responses.

3 Background

Reinforcement Learning from Human Feedback (RLHF) serves as a pivotal method for aligning LLMs with human preferences, particularly in terms of their helpfulness and harmlessness (Ouyang et al., 2022). RLHF begins with learning a latent reward model $r(\boldsymbol{x}, \boldsymbol{y})$ that can implicitly capture human preferences for pairwise comparisons, which are often nuanced or subjective to be explicitly defined (Ouyang et al., 2022). Specifically, given a collection of human preference data $\mathcal{D}_{\text{p}} = \{(\boldsymbol{x}, \boldsymbol{y}^w, \boldsymbol{y}^l)\}$, where $\boldsymbol{x}$ is a user input prompt, $\boldsymbol{y}^w, \boldsymbol{y}^l$ are the preferred (chosen) and non-preferred (rejected) responses, a reward model is usually optimized by minimizing a ranking loss following the Bradley-Terry (Bradley and Terry, 1952) objective:

$$-\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}^w,\boldsymbol{y}^l)\sim\mathcal{D}_{\text{p}}}\Big[\log\sigma(r(\boldsymbol{x},\boldsymbol{y}^w) - r(\boldsymbol{x},\boldsymbol{y}^l))\Big], \tag{1}$$

where $\sigma(\cdot)$ is the Sigmoid function. Intuitively, equation 1 induces $r(\boldsymbol{x}, \boldsymbol{y})$ to assign a higher reward score to the preferred pairs $(\boldsymbol{x}, \boldsymbol{y}^w)$ than the rejected response $(\boldsymbol{x}, \boldsymbol{y}^l)$ with respect to an input $\boldsymbol{x}$. Therefore, the optimized reward model serves as a proxy for human preferences, enabling the subsequent RL fine-tuning phase. While effective, the vanilla equation 1 objective also suffers from the lack of sufficient distinction between similar responses (Wang et al., 2024a; Touvron et al., 2023), especially when faced with ambiguous training samples. With a learned RM $r(\boldsymbol{x}, \boldsymbol{y})$, RLHF optimizes the target LLM policy $\pi_\theta(\boldsymbol{y}|\boldsymbol{x})$ for each input $\boldsymbol{x}$ by maximizing $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\boldsymbol{y}\sim\pi_\theta(\boldsymbol{y}|\boldsymbol{x})}[r(\boldsymbol{x}, \boldsymbol{y}) - \text{KL}(\pi_\theta(\boldsymbol{y}|\boldsymbol{x})\|\pi_{\text{ref}}(\boldsymbol{y}|\boldsymbol{x}))]$. To solve the RLHF objective, Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Group Reward Proxy Optimization (GRPO) (Shao et al., 2024) have been recognized as the mainstream optimization algorithms (Rafailov et al., 2024). Recently, several simplified alignment methods have been proposed to avoid the significant generation cost required by online RLHF methods, such as Rafailov et al. (2024); Du et al. (2025). Beyond aligning with human preferences, the paradigm of RLHF has also successfully expanded into other NLP tasks, such as prompt refinement (Li et al., 2025a) and LLM safety detection (Du et al., 2024).

Optimal Transport (OT) is a popular measurement for comparing distributions (Peyré et al., 2019), which has been successfully applied on various machine learning tasks (Li et al., 2025c; Guo et al., 2022; Gao et al., 2023a). We mainly consider the discrete form in this manuscript. Given two sets of points, their discrete distributions can be formulated as $P = \sum_{n=1}^{N} u_n \delta_{x_n}$ and $Q = \sum_{m=1}^{M} v_m \delta_{y_m}$, where δ is the Dirac function, and $\boldsymbol{u}$ and $\boldsymbol{v}$ are probability distributions summing to 1. The OT distance between P and Q can be measured as:

$$\min_{\mathbf{T}\in\Pi(P,Q)} \langle \mathbf{T}, \mathbf{C} \rangle = \sum_{n}^{N}\sum_{m}^{M} T_{nm} C_{nm}, \tag{2}$$

where $\mathbf{C} \in \mathbb{R}_{>0}^{n \times m}$ is the cost matrix (e.g., cosine distance) whose each element denotes the distance between x_n and y_m, and the transport probability matrix $\mathbf{T} \in \mathbb{R}_{>0}^{N \times M}$ satisfies:

$$\Pi(P, Q) := \left\{ \mathbf{T} | \sum_{n=1}^{N} T_{nm} = v_m, \sum_{m=1}^{M} T_{nm} = u_n \right\}. \tag{3}$$

As directly optimizing equation 2 can be computationally expensive, an entropic constraint $H(\mathbf{T}) = -\sum_{nm} T_{nm} \ln T_{nm}$ is often introduced for faster optimization (Cuturi, 2013).

4 Method

This section presents our adaptive margin estimation method for robust reward modeling, formulated from a distribution-aware perspective that can be solved by Optimal Transport. Our core idea is to dynamically adjust the margin for each training triplet based on their inherent semantic similarity and model-predicted reward difference, ensuring that the model focuses more on difficult samples while avoiding over-fitting on easy ones.

Motivation. Intuitively, a reasonable margin μ_i should reflect the difficulty of distinguishing between a preferred $\boldsymbol{y}_i^w$ and non-preferred response $\boldsymbol{y}_i^l$ for an input prompt $\boldsymbol{x}_i$. Specifically, the margin μ_i should be larger for samples with high semantic similarity but low reward difference, indicating that the model finds it challenging to differentiate between them. Conversely, it should be smaller for samples with high reward differences, where the model already demonstrates a clear preference, thereby reducing over-fitting.

Reward Margin Formulation. Given a set of preference triplets $\{(\boldsymbol{x}_i, \boldsymbol{y}_i^w, \boldsymbol{y}_i^l)\}_{i=1}^N$, for arbitrary two pairs of preference $(\boldsymbol{x}_i, \boldsymbol{y}_i^w)$ and $(\boldsymbol{x}_j, \boldsymbol{y}_j^l)$, we define the corresponding predicted reward difference as $\Delta f_{ij} = r(\boldsymbol{x}_i, \boldsymbol{y}_i^w) - r(\boldsymbol{x}_j, \boldsymbol{y}_j^l)$, and the semantic similarity between $(\boldsymbol{x}_i, \boldsymbol{y}_i^w)$ and $(\boldsymbol{x}_j, \boldsymbol{y}_j^l)$ as $S_{ij} = S\left((\boldsymbol{x}_i, \boldsymbol{y}_i^w), (\boldsymbol{x}_j, \boldsymbol{y}_j^l)\right)$, where $S(\cdot, \cdot)$ is a measure of semantic similarity for the input (e.g., cosine similarity). To estimate the adaptive margins μ for these (preferred, non-preferred) pairs, we first build two discrete probability distributions P and Q as follows:

$$P = \sum_{i=1}^{N} \frac{1}{N} \delta_{(\boldsymbol{x}_i, \boldsymbol{y}_i^w)}, \quad Q = \sum_{j=1}^{N} \frac{1}{N} \delta_{(\boldsymbol{x}_j, \boldsymbol{y}_j^l)}, \tag{4}$$

where N indicates the number of training triples. Therefore, we can define the margins by OT distance:

$$\mathrm{OT}(P, Q) = \min_{\mathbf{T} \in \Pi(P,Q)} \langle \mathbf{T}, \mathbf{C} \rangle - \beta H(\mathbf{T}), \tag{5}$$

where β is a hyper-parameter for the entropy constraint $H(\mathbf{T})$ and the C_{ij} measures the distance between $(\boldsymbol{x}_i, \boldsymbol{y}_i^w)$ and $(\boldsymbol{x}_j, \boldsymbol{y}_j^l)$. The transport plan $\mathbf{T}$ satisfies $\Pi(P, Q) = \left\{ \mathbf{T} \in \mathbb{R}_+^{N \times N} | \sum_{i=1}^N T_{ij} = N_j, \ \sum_{j=1}^N T_{ij} = N_i \right\}$. This formulation allows us to capture the distributional differences between preferred and non-preferred responses in a principled manner.

Cost Matrix Design. Cost matrix acts as a determinant in the optimization of the transport plan between P and Q. Although it is possible to use point-wise distances like cosine metric, these only focus on semantic similarity and ignore the reward differences, leading to suboptimal margin estimation. Recall our motivation that the margin should reflect both the semantic similarity and the model's predicted reward differences to capture the true difficulty of distinguishing between preferred and non-preferred responses. Therefore, we design the cost matrix $\mathbf{C}$ to incorporate both semantic similarity $S\left((\boldsymbol{x}_i, \boldsymbol{y}_i^w), (\boldsymbol{x}_j, \boldsymbol{y}_j^l)\right)$ and reward differences $\Delta f_i = r(\boldsymbol{x}_i, \boldsymbol{y}_i^w) - r(\boldsymbol{x}_j, \boldsymbol{y}_j^l)$ by:

$$C_{ij} = \underbrace{\gamma \cdot S_{i,j}}_{\text{Semantic Similarity}} + \underbrace{(1-\gamma) \cdot (1 - \sigma(\Delta f_{ij})}_{\text{Reward Differences}} \tag{6}$$

where γ is a balance hyper-parameter and σ is Sigmoid function. We use cosine similarity for $S(\cdot, \cdot)$. Clearly, larger semantic similarity leads to a significant increase in cost, while larger reward differences only bring a slight improvement, aligning with our motivation. As a result, this formulation adaptively captures the learning differences between preferred and non-preferred responses. For semantic measurement, we extract the last hidden state of the last non-pad token in each input pair.

Distributional Adaptive Margin Estimation and Training Loss. Using this carefully designed cost matrix $\mathbf{C}$, we can compute the optimal transport plan $\mathbf{T}^*$ by equation 5. As a result, the adaptive margin μ_i for the i-th triplet is then derived from the $\mathbf{T}^*$ and $\mathbf{C}$, which we name "APLOT":

$$\mu_i = \sum_{j=1}^{N} T_{ij}^* \cdot C_{ij}. \tag{7}$$

This formulation ensures that the margin for each triplet is influenced by its pairwise relationships

Algorithm 1: Reward Modeling with Adaptive Margin Estimation in Mini-Batch.

Input : Preference Dataset $D_p = \{(\boldsymbol{x}_i, \boldsymbol{y}_i^w, \boldsymbol{y}_i^l)\}_{i=1}^N$, hyper-parameters γ.
Output : Trained reward model r

Initialize a reward model r with parameters θ;
while *not converged* **do**
  Sample a mini-batch of triplets $\{(\boldsymbol{x}_i, \boldsymbol{y}_i^w, \boldsymbol{y}_i^l)\}_{i=1}^B \sim D_p$;
  Compute predicted reward differences $\Delta f_{ij} = r(\boldsymbol{x}_i, \boldsymbol{y}_i^w) - r(\boldsymbol{x}_j, \boldsymbol{y}_j^l)$;
  Calculate semantic similarities $S_{ij} = S\left((\boldsymbol{x}_i, \boldsymbol{y}_i^w), (\boldsymbol{x}_j, \boldsymbol{y}_j^l)\right)$ using cosine similarity;
  Construct cost matrix $\mathbf{C}$ by $\mathbf{C}_{ij} = \gamma \cdot S_{ij} + (1-\gamma) \cdot (1 - \sigma(\Delta f_{ij}))$;
  Build P and Q by $P = \sum_{i=1}^N \frac{1}{N} \delta_{(\boldsymbol{x}_i, \boldsymbol{y}_i^w)}$, $Q = \sum_{j=1}^N \frac{1}{N} \delta_{(\boldsymbol{x}_j, \boldsymbol{y}_j^l)}$;
  Solve the OT problem to obtain the optimal transport plan $\mathbf{T}^*$ by $\mathbf{T}^* = \underset{\mathbf{T} \in \Pi(P,Q)}{\arg\min} \langle \mathbf{T}, \mathbf{C} \rangle - 0.1 \times H(\mathbf{T})$;
  Estimate adaptive margins $\mu_i = \sum_{j=1}^B T_{ij}^* \cdot C_{ij}$ for each triplet in the mini-batch;
  Compute the adjusted Ranking Loss $\mathcal{L}$ using the estimated adaptive margins:
  $\mathcal{L} = -\frac{1}{B} \sum_{i=1}^B \log \sigma \left(r(\boldsymbol{x}_i, \boldsymbol{y}_i^w) - r(\boldsymbol{x}_i, \boldsymbol{y}_i^l) - \mu_i\right)$;
  Update the reward model parameters θ using gradient descent with the computed loss $\mathcal{L}$;
end

with other triplets, as captured by the transport plan. Triplets that are more challenging to distinguish will receive larger margins, while easier triplets will receive smaller margins. Finally, we can incorporate our adaptive margin to BT ranking loss equation 1 to optimize a robust reward model by minimizing the following objective:

$$-\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}^w,\boldsymbol{y}^l)\sim\mathcal{D}_\text{p}}\Big[\log\sigma(r(\boldsymbol{x},\boldsymbol{y}^w)-r(\boldsymbol{x},\boldsymbol{y}^l)-\mu)\Big]. \tag{8}$$

In addition, we introduce a simpler baseline of our method by estimating the margin μ_i in a point-to-point way, which we name "PointMargin":

$$\mu_i = \sum_{j=1}^{N} C_{ij}. \tag{9}$$

The main difference between APLOT (equation 7) and PointMargin (equation 9) lies in the approach to aggregating cost information contained in the matrix $\mathbf{C}$, where PointMargin computes the margin as a simple summation of costs, providing a straightforward but local, point-to-point baseline with less training consumption. APLOT, in contrast, takes a more sophisticated distributional approach, leveraging the optimal transport plan $\mathbf{T}$ to derive a margin that is a weighted sum of costs, thereby reflecting a globally optimal matching process across the distributions of preferred and dispreferred responses with additional computation.

Why Margin can help RM better differentiate the responses? The introduction of a margin μ enhances the RM's ability to differentiate between preferred and dispreferred responses given an input $\boldsymbol{x}$. Minimizing equation 8 encourages the score difference $= r(\boldsymbol{x}, \boldsymbol{y}^w) - r(\boldsymbol{x}, \boldsymbol{y}^l)$ to exceed μ, thereby enforcing a more pronounced separation between preferred and dispreferred responses. When $s \leq \mu$, the model receives stronger gradient signals (e.g., gradient magnitude $1 - (s - \mu) \geq 0.5$), guiding it to focus on harder samples with small but positive margins. A larger margin μ naturally correlates with higher difficulty, prompting the RM to concentrate on finer-grained distinctions and improve learning where it matters most. Our method further benefits from adaptively estimating μ, allowing the learning objective to dynamically adjust based on the difficulty of sample pairs. More gradient analysis is provided in Appendix A.

In summary, by formulating the adaptive margin estimation as an Optimal Transport problem, we gain a principled way to incorporate both semantic similarity and predicted reward differences into the margin. This distribution-aware method adjusts μ_i based on the specific characteristics of each triplet, providing a robust defense against over-fitting while directing the model's attention towards the most challenging cases. This approach ultimately improves the model's performance in reward modeling tasks by ensuring consistent and robust reward assignments. We summarize the training algorithm in Alg. 1.

5 Experiment

5.1 Setup

Datasets. By following Yang et al. (2024b), we leverage the Unified-Feedback (UF) preference dataset[2] to demonstrate the effectiveness of our

[2] https://huggingface.co/datasets/llm-blender/Unified-Feedback

Table 1: Results on ID and OOD evaluation with 40K training data from Unified Feedback based on gemma-2b-it. The best performance in each task is **bold**. Baseline results are cited from Yang et al. (2024b). We set HardMargin as 1.0.

Method	Unified Feedback	HHH Alignment	MT Bench	RewardBench				
				Avg.	Chat	Chat-Hard	Safety	Reasoning
BT - Vanilla	68.8	70.3	69.1	64.5	95.8	37.3	59.9	64.8
BT - HardMargin	69.6	69.8	71.0	66.1	97.2	37.5	56.8	72.7
BT - LabelSmooth	68.5	68.8	71.9	61.1	91.6	39.0	53.8	60.2
BT - Ensemble	69.9	72.2	71.1	65.2	96.1	38.2	58.8	67.6
GRM + DPO	70.2	71.6	71.3	70.8	97.8	42.1	77.9	65.2
GRM + DPO-NoRef	71.4	76.6	72.1	66.6	92.5	39.9	72.5	61.4
GRM + SFT	71.5	78.7	73.0	66.8	94.1	41.9	69.5	61.5
APLOT (OURS)	**73.84**	**81.25**	**75.23**	**74.10**	**97.21**	**42.54**	**80.81**	**72.17**

method when compared with vanilla BT training objective. For a comprehensive evaluation of the reward model's performance, we consider both in-distribution (ID) and out-of-distribution (OOD). Specifically, not only are models evaluated on the standard 1K UF eval set (ID), but we also compare the performance of different RM methods on three popular OOD datasets: HHH-Alignment (Coste et al., 2023), MT-Bench Human Judgments (Zheng et al., 2023), RewardBench (Lambert et al.) and RM-Bench (Liu et al., 2024c). The HHH-Alignment dataset mainly evaluates a language model from the perspectives of helpfulness, honesty, and harmlessness. MT-Bench contains 3.3K expert-level pairwise human preferences for model responses generated by 6 models in response to MT-bench questions. Besides, RewardBench is a popular benchmark designed to comprehensively evaluate the capabilities and safety of reward models. On the other hand, we also adopt the Skywork Reward Preference (SP) dataset (Liu et al., 2024a) for reward model training when compared with several SOTA RM methods.

Base Models and Training Details. For base models, we adopt gemma-2b-it (Team et al., 2024) and Llama-3.1-8B-Instruct (Dubey et al., 2024). Training details can be found in Appendix B.

Baselines. We evaluate the performance of our method with several baselines, including (1) Vanilla BT reward model; (2) BT-Variants, including BT w/ margin, label smooth, PosReg and ensemble (Touvron et al., 2023; Wang et al., 2024a); (4) GRM (Yang et al., 2024b) that designs to regularize the hidden states by incorporating a DPO loss (Rafailov et al., 2024) and its variants.

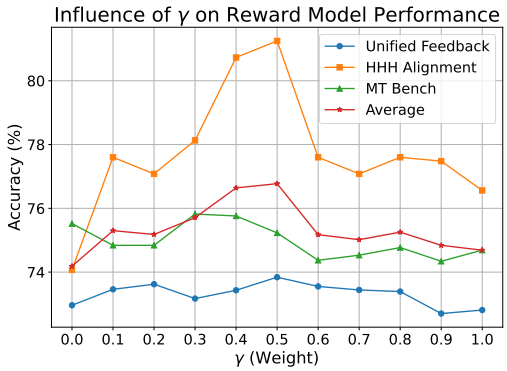

Figure 2: Influence of the weight γ on our reward model's performance across different tasks, which balances the semantic consistency and reward difference.

5.2 Results and Analysis

Performance on ID and OOD settings compared with baselines. By following Yang et al. (2024b), we first randomly sample 40K samples from the UF dataset for RM training with the help of LoRA (Hu et al., 2021). As shown in Table 1, our method consistently outperforms all baseline methods on both ID and OOD evaluation tasks. Specifically, our method achieves the highest scores of 73.84 on Unified Feedback, 81.25 on HHH Alignment, and 75.23 on MT Bench, indicating that our method significantly improves the model's ability to generalize to both ID and OOD reward evaluation datasets. The strong performance on OOD tasks, in particular, demonstrates the robustness of our approach in handling unseen scenarios. Compared to the baseline methods, our approach shows a clear advantage, especially in scenarios where the model needs to differentiate between similar responses with subtle reward differences.

Influence Analysis of γ. The hyperparameter γ balances the importance of semantic similarity and

Table 2: Performance comparison of different reward models on RewardBench. The best performance in each task is in **bold** and we cite results from Liu et al. (2024a).

Type	Method	Avg.	Chat	Chat-Hard	Safety	Reasoning
Generative	SFR-LLaMa-3.1-70B-Judge-I	92.7	96.9	84.8	91.6	**97.6**
	Gemini-1.5	86.8	94.1	77.0	85.8	90.2
	GPT-4o	86.7	96.1	76.1	88.1	86.6
	SFR-nemo-12B-Judge-r	90.3	97.2	82.2	86.5	95.1
Discriminative	Nemotron-340B-Reward	92.2	95.8	87.1	92.2	93.6
	ArmoRM-Llama3-8B-v0.1	90.8	96.9	76.8	92.2	97.3
	InternLM-20B-Reward	90.2	**98.9**	76.5	89.9	95.8
	Llama-3-OffsetBias-RM-8B	89.4	97.2	81.8	86.8	91.9
	Llama-3.1-BT-RM-8B	91.8	94.6	84.5	91.5	96.5
	Skywork-Reward-Llama-3.1-8B	92.5	95.8	87.3	90.6	96.2
	APLOT-Scratch-Llama-3.1-8B	92.1	97.2	84.9	92.1	94.2
	APLOT-Skywork-Llama-3.1-8B	**94.4**	93.9	**89.0**	**93.2**	97.4

Table 3: Performance comparison of different reward models on RM-Bench. The best performance in each task is in **bold** and we cite results from Liu et al. (2024c).

Type	Method	Avg.	Chat	Math	Code	Safety
DPO	upstage/SOLAR-10.7B-Instruct-v1.0	64.8	**78.6**	52.3	49.6	78.9
	allenai/tulu-2-dpo-13b	63.8	66.4	51.4	51.8	85.4
Discriminative	URM-LLaMa-3.1-8B	70.0	71.2	61.8	54.1	93.1
	Nemotron-340B-Reward	69.5	71.2	59.8	59.4	87.5
	Llama-3-OffsetBias-RM-8B	69.0	71.3	61.9	53.2	89.6
	internlm2-20b-reward	68.3	63.1	**66.8**	56.7	86.5
	GRM-llama-3-8B-sftreg	68.2	62.7	62.5	57.8	90.0
	Skywork-Reward-Llama-3.1-8B	70.1	69.5	60.6	**54.5**	95.7
	APLOT-Scratch-Llama-3.1-8B	71.68	72.44	63.58	54.19	96.32
	APLOT-Skywork-Llama-3.1-8B	**72.10**	72.82	63.89	54.24	**96.50**

reward difference in the cost matrix. To quantitatively evaluate the influence of γ, we train RMs on the a 40K subset of UF dataset based on gemma-2b-it equipped with Lora. As shown in Figure 2, optimal performance across tasks is achieved when γ is around 0.4 to 0.6, with $\gamma = 0.5$ consistently leading to the highest accuracy in three test datasets, as well as the overall average performance. When γ is too high or too low, the model's performance decreases, as it causes the model to disproportionately emphasize one aspect over the other. This indicates that a balanced γ is crucial for our approach to effectively incorporate both semantic similarity and reward difference into margin estimation, thereby enhancing the reward model's performance. Our method performs best when γ is set around 0.5, highlighting the importance of balancing these two factors in our cost matrix design, specifically for OOD settings.

Performance on RewardBench and RM-Bench. Table 2 presents a comprehensive evaluation of various reward models on the RewardBench dataset, demonstrating the effectiveness of our proposed method. We train our RM on SP with full parameter tuning. Notably, our method achieves compelling results even when trained from scratch. Specifically, "APLOT-Scratch-Llama-3.1-8B" (trained scratch from Llama-3.1-8B-Instruct) attains a strong average score of 92.1, comparable to other high-performing reward models and even better than several RM with higher scale, highlighting the inherent effectiveness of our approach. Furthermore, our method exhibits remarkable flexibility by also serving as a powerful tool to enhance pre-existing reward models. By applying our technique to refine the already strong Skywork-Reward-Llama-3.1-8B model, we achieve a further performance boost, reaching a best score of 94.4 with "APLOT-Skywork-Llama-3.1-8B" and signif-

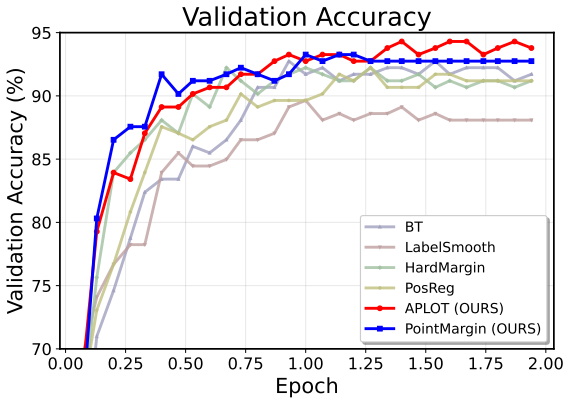

Figure 3: Illustration of the convergence and performance of our proposed method, in terms of validation accuracy over training epochs. Both APLOT and PointMargin demonstrate faster convergence, achieving higher accuracy with fewer epochs, and ultimately reach better accuracies compared to baselines.

icantly improving results in key areas like "Chat-Hard" (89.0) and "Safety" (93.2). While generative models like SFR-LLaMa-3.1-70B-Judge-I excel in specific tasks such as "Reasoning" (97.6), our discriminative approach demonstrates both intrinsic efficacy and the ability to enhance other models, showcasing its versatility. In conclusion, these findings underscore the dual advantage of our reward modeling method: strong performance on its own and the capacity to effectively "plug-and-play" to elevate the performance of existing reward models. Besides, we also evaluate the performance of our methods on RM-Bench, which is another popular and more challenging RM benchmark. As shown in Table 3, we also observe that our method brings better performance compared with several strong baselines.

Convergence and Validation Accuracy Comparison. We evaluate the convergence speed compared with baseline methods. Experimentally, we randomly sampled 40K training points from SP dataset. Figure 3 depicting validation accuracy against training epoch, clearly demonstrates the superior convergence and performance of our proposed margin estimators APLOT (equation 7) and PointMargin (equation 9). Experimentally, both APLOT (red line with circles) and PointMargin (blue line with squares) exhibit significant convergence speed learning and better performance by achieving >90% validation accuracy before 0.5 epoch, compared to baselines that struggle to reach similar accuracy even after 1.0 epoch. This indicates that our methods learn more efficiently and require less training to reach optimal performance. In terms of final performance, PointMargin shows substantial improvement over baselines, while APLOT further enhances performance, surpassing 94% accuracy after nearly 2.0 epochs, where baselines plateau in the 88%-92% range. In conclusion, these results underscore the effectiveness of our proposed techniques, with PointMargin providing a strong improvement and APLOT's enhanced performance demonstrating the value of a distributional perspective in learning more robust reward models.

Table 4: Performance comparison of RMs on the Unified Feedback, HHH Alignment, and MT Bench datasets under 20% label noise within the training dataset SP.

Method	Unified Feedback	HHH Alignment	MT Bench	Avg.
BT - Vanilla	71.53	72.92	72.68	72.38
BT - HardMargin	70.43	77.60	73.85	73.96
BT - LabelSmooth	70.62	76.56	71.28	72.82
BT - PosReg	71.08	79.17	**75.11**	75.21
APLOT (OURS)	**71.82**	**81.78**	74.88	**76.16**

Performance against Label-Noise Label noise are inevitable during human preference annotations (Wang et al., 2024a), which hinders the generalization and effectiveness of reward models. To evaluate the robustness of our method, we design to randomly assign 20% label noise into a 20K SP training subset. As shown in Table 4, we find that several variants bring improvements to the vanilla BT RM, while our method can significantly enhance the reward model by more accurately judging the sample quality, even within the noisy training annotations.

5.3 Evaluation on RLHF

Best-of-N (BoN) Sampling Test. Figure 4 reports BoN results for the Qwen 2.5-3 B and 7 B-Instruct models. For each setting, we train proxy reward models on a randomly sampled 40 K subset of the SP corpus using Llama-3.1-8B-Instruct as the backbone. Following Coste et al. (2024); Gao et al. (2023b), we generate N candidate completions for every prompt in a 300-instance out-of-distribution (OOD) test set, rank them with the proxy RMs, and then evaluate the chosen responses with a high-fidelity gold reward model (Skywork-Reward-Gemma-2-27 B). The average gold score over the 300 prompts thus reflects the true quality of the proxy-selected answers. We vary the KL-divergence budget from 0 to 5, which—through the relation $\mathrm{KL}_{\mathrm{Bo}N} = \log N - \frac{N-1}{N}$ (Gao et al.,

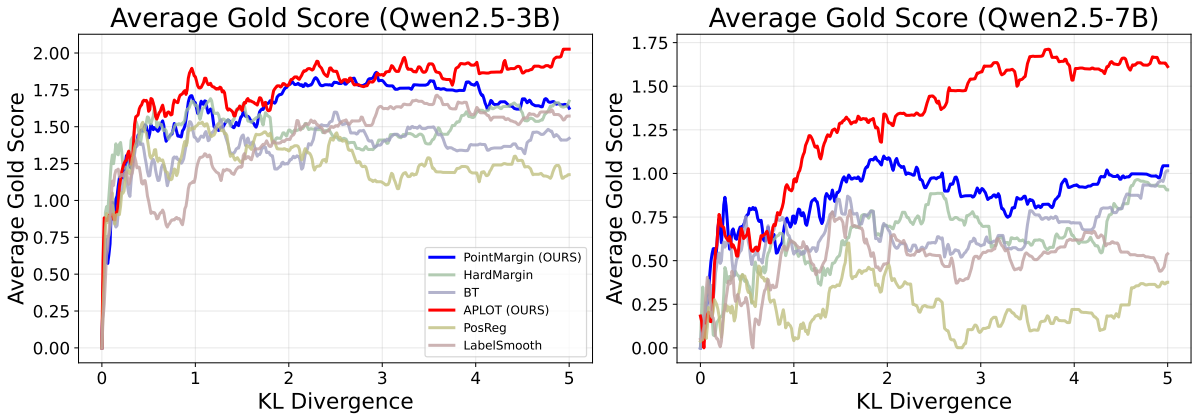

Figure 4: Gold scores from the Best-of-N (BoN) test, using responses sampled from Qwen2.5-3B and 7B-Instruct, respectively. Rewards are normalized to start at 0. APLOT shows robust alignment with gold rewards despite increasing KL Divergence.

2023b)—corresponds to N ranging from 1 to 405. A well-behaved proxy RM should yield monotonically increasing proxy and gold scores as the KL budget (and hence N) grows. Several baselines instead plateau or even decline once KL > 2 (see the right-hand plot for Qwen 2.5-7B-Instruct), revealing over-optimization. In contrast, our APLOT approach maintains a steady rise in gold score across the entire KL range, demonstrating its ability to curb over-optimization and underscoring APLOT's robustness as a proxy reward model for RLHF.

PPO. Beyond the BoN Test, we conduct the PPO experiments to practically investigate whether our reward model helps better RLHF training with the help of an adaptive margin. As shown in Table 5, our experimental results highlight the better performance of the policy model fine-tuned with our APLOT. On the OpenRLHF-Llama3-8B-SFT base, our APLOT model achieve an average of 58.55, compared to 55.68 for the baseline and 56.94 for SKRM. These findings validate the effectiveness of our APLOT reward model in consistently enhancing the capabilities of language models through PPO fine-tuning. Training and evaluation details can be found in Appendix C.

6 Conclusion

In this work, we proposed to enhance the pairwise preference reward model with a novel adaptive margin to achieve better generalization in RLHF. Our approach leverages Optimal Transport (OT) to dynamically adjust the margin based on semantic similarity and model-predicted reward difference, ensuring that the model focuses more on challenging samples while avoiding over-fitting on easier ones.

Table 5: Benchmark Evaluation. Baseline indicates the benchmark performance on vanilla OpenRLHF-Llama3-8B-SFT, respectively. SWRM is that of the policy model trained based on the reward model Skywork-RM-Llama3.1-8B-Instruct, and OURS is that based on our APLOT-RM-Llama3.1-8B-Instruct.

Benchmark	Baseline	SKRM	APLOT
$\text{GSM8K}_{\text{acc}}$	74.83	78.17	**79.23**
$\text{Hellaswag}_{\text{acc}}$	72.51	74.76	**76.12**
$\text{IFeval}_{\text{acc}}$	44.92	45.10	**48.98**
MMLU_{acc}	54.45	52.40	**55.62**
$\text{ProcessBench}_{\text{acc}}$	4.46	10.31	**10.62**
Race_{acc}	79.21	78.82	**80.93**
BBH_{acc}	61.20	**62.68**	62.42
$\text{Humaneval}_{\text{pass@1}}$	60.98	57.32	**63.41**
$\text{TriviaQA}_{\text{acc}}$	48.53	**52.86**	49.63
Avg.	55.68	56.94	**58.55**

Through extensive experiments, we have demonstrated that our method consistently outperforms existing reward modeling techniques across multiple benchmarks, showing significant improvements in both in-distribution and out-of-distribution tasks. The ablation study further highlights the importance of balancing semantic similarity and reward difference in the cost matrix design.

7 Acknowledgments

This work was supported by the Guangxi Key R&D Project (No. AB24010167), the Project (No. 20232ABC03A25), and the Futian Healthcare Research Project (No.FTWS002). This work was also supported by the National Natural Science Foundation of China (NSFC) under Grant 62306125 and Grant 62402158.

8 Limitation and Future Work

Despite the promising results demonstrated by our method, there are several limitations that need to be acknowledged. Firstly, our current method is designed for language-based reward models and has not been adapted for multi-modal inputs or progress reward models, and cannot be explicitly adapted to generative reward modeling. The complexity of multi-modal data and the dynamic nature of progress reward modeling pose additional challenges that our current approach does not address. Secondly, our method relies on the quality and representativeness of the training data. If the training data is biased or lacks diversity, it may limit the performance of our method.

In future work, we plan to address the limitations mentioned above. We will explore the application of our method in multi-modal reward modeling, where the reward function needs to consider both text and other modalities such as images or audio. This extension will require the development of new techniques to effectively integrate multi-modal information into the reward model. Additionally, we aim to investigate the potential of our method in progress and generative reward modeling, where the reward function needs to adapt to the progress of the learning process. This will involve designing new algorithms that can dynamically adjust the reward function based on the agent's progress.

References

Pengcheng He Michel Galley Jianfeng Gao Baolin Peng, Chunyuan Li. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

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.

Alex J Chan, Hao Sun, Samuel Holt, and Mihaela van der Schaar. 2024. Dense reward for free in reinforcement learning from human feedback. *arXiv preprint arXiv:2402.00782*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021a. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021b. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. 2023. Reward model ensembles help mitigate overoptimization. *arXiv preprint arXiv:2310.02743*.

Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. 2024. Reward model ensembles help mitigate overoptimization. *Preprint*, arXiv:2310.02743.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26.

Chongyuan Dai, Jinpeng Hu, Hongchang Shi, Zhuo Li, Xun Yang, and Meng Wang. 2025. Psyche-r1: Towards reliable psychological llms through unified empathy, expertise, and reasoning. *arXiv preprint arXiv:2508.10848*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Yuhao Du, Zhuo Li, Pengyu Cheng, Zhihong Chen, Yuejiao Xie, Xiang Wan, and Anningzhe Gao. 2025. Simplify rlhf as reward-weighted sft: A variational method. *Preprint*, arXiv:2502.11026.

Yuhao Du, Zhuo Li, Pengyu Cheng, Xiang Wan, and Anningzhe Gao. 2024. Detecting ai flaws: Target-driven attacks on internal faults in language models. *Preprint*, arXiv:2408.14853.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak

Ramachandran, and 1 others. 2023. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint arXiv:2312.09244*.

Jintong Gao, He Zhao, Zhuo Li, and Dandan Guo. 2023a. Enhancing minority classes by mixing: An adaptative optimal transport approach for long-tailed classification. In *Advances in Neural Information Processing Systems*, volume 36, pages 60329–60348. Curran Associates, Inc.

Leo Gao, John Schulman, and Jacob Hilton. 2023b. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR.

Dandan Guo, Zhuo Li, meixi zheng, He Zhao, Mingyuan Zhou, and Hongyuan Zha. 2022. Learning to re-weight examples with optimal transport for imbalanced classification. In *Advances in Neural Information Processing Systems*, volume 35, pages 25517–25530. Curran Associates, Inc.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Jinpeng Hu, Tengteng Dong, Luo Gang, Hui Ma, Peng Zou, Xiao Sun, Dan Guo, Xun Yang, and Meng Wang. 2024. Psycollm: Enhancing llm for psychological understanding and evaluation. *IEEE Transactions on Computational Social Systems*.

Jinpeng Hu, DanDan Guo, Yang Liu, Zhuo Li, Zhihong Chen, Xiang Wan, and Tsung-Hui Chang. 2023. A simple yet effective subsequence-enhanced approach for cross-domain ner. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12890–12898.

Jinpeng Hu, Zhuo Li, Zhihong Chen, Zhen Li, Xiang Wan, and Tsung-Hui Chang. 2022. Graph enhanced contrastive learning for radiology findings summarization. *Preprint*, arXiv:2204.00203.

Jinpeng Hu, Hongchang Shi, Chongyuan Dai, Zhuo Li, Peipei Song, and Meng Wang. 2025a. Beyond emotion recognition: A multi-turn multimodal emotion understanding and reasoning benchmark. *arXiv preprint arXiv:2508.16859*.

Jinpeng Hu, Ao Wang, Qianqian Xie, Hui Ma, Zhuo Li, and Dan Guo. 2025b. Agentmental: An interactive multi-agent framework for explainable and adaptive mental health assessment. *Preprint*, arXiv:2508.11567.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *Preprint*, arXiv:1705.03551.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *Preprint*, arXiv:1704.04683.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, and 1 others. Rewardbench: Evaluating reward models for language modeling, march 2024. *URL http://arxiv. org/abs/2403.13787*.

Zhuo Li, Yuhao Du, Jinpeng Hu, Xiang Wan, and Anningzhe Gao. 2025a. Self-instructed derived prompt generation meets in-context learning: Unlocking new potential of black-box LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1840–1857, Vienna, Austria. Association for Computational Linguistics.

Zhuo Li, Yuhao Du, Xiaoqi Jiao, Yiwen Guo, Yuege Feng, Xiang Wan, Anningzhe Gao, and Jinpeng Hu. 2025b. Add-one-in: Incremental sample selection for large language models via a choice-based greedy paradigm. *Preprint*, arXiv:2503.02359.

Zhuo Li, He Zhao, Anningzhe Gao, Dandan Guo, Tsung-Hui Chang, and Xiang Wan. 2025c. Prototype-oriented clean subset extraction for noisy long-tailed classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(8):7953–7965.

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

Tianqi Liu, Wei Xiong, Jie Ren, Lichang Chen, Junru Wu, Rishabh Joshi, Yang Gao, Jiaming Shen, Zhen Qin, Tianhe Yu, Daniel Sohn, Anastasiia Makarova, Jeremiah Liu, Yuan Liu, Bilal Piot, Abe Ittycheriah, Aviral Kumar, and Mohammad Saleh. 2024b. Rrm: Robust reward model training mitigates reward hacking. *Preprint*, arXiv:2409.13156.

Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. 2024c. Rm-bench: Benchmarking reward models of language models with subtlety and style. *Preprint*, arXiv:2410.16184.

R Duncan Luce. 1959. *Individual choice behavior*, volume 4. Wiley New York.

OpenAI. 2023. ChatGPT, Mar 14 version. https://chat.openai.com/chat.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Gabriel Peyré, Marco Cuturi, and 1 others. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

Robin L Plackett. 1975. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202.

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.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, and 1 others. 2022a. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022b. Challenging big-bench tasks and whether chain-of-thought can solve them. *Preprint*, arXiv:2210.09261.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, and 1 others. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, Songyang Gao, Nuo Xu, Yuhao Zhou, Xiaoran Fan, Zhiheng Xi, Jun Zhao, Xiao Wang, Tao Ji, Hang Yan, and 8 others. 2024a. Secrets of rlhf in large language models part ii: Reward modeling. *Preprint*, arXiv:2401.06080.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024b. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *Preprint*, arXiv:2406.12845.

Peifeng Wang, Austin Xu, Yilun Zhou, Caiming Xiong, and Shafiq Joty. 2024c. Direct judgement preference optimization. *Preprint*, arXiv:2409.14664.

Tianlu Wang, Ilia Kulikov, Olga Golovneva, Ping Yu, Weizhe Yuan, Jane Dwivedi-Yu, Richard Yuanzhe Pang, Maryam Fazel-Zarandi, Jason Weston, and Xian Li. 2024d. Self-taught evaluators. *arXiv preprint arXiv:2408.02666*.

Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. 2017. Sampling matters in deep embedding learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2840–2848.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024a. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. 2024b. Regularizing hidden states enables learning generalizable reward model for llms. *arXiv preprint arXiv:2406.10216*.

Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. 2024c. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. *Preprint*, arXiv:2402.10207.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *Preprint*, arXiv:1905.07830.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. Processbench: Identifying process errors in mathematical reasoning. *Preprint*, arXiv:2412.06559.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

A Analysis of Margin μ's Impact on Loss and Gradient

Let the per-sample loss for a triplet $(\boldsymbol{x}, \boldsymbol{y}^w, \boldsymbol{y}^l)$ be defined as:

$$\ell = -\log(\sigma(s-\mu)), \quad s = r(\boldsymbol{x}, \boldsymbol{y}^w) - r(\boldsymbol{x}, \boldsymbol{y}^l), \quad \mu > 0 \tag{10}$$

where $r(\boldsymbol{x}, \boldsymbol{y})$ denotes the model score for input-output pair $(\boldsymbol{x}, \boldsymbol{y})$.

Impact on Loss Value. The sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ is monotonically increasing. The negative logarithm $-\log(\cdot)$ is monotonically decreasing.

1. If $s \leq \mu$ (Difficult/Marginal Sample), we will have $\sigma(s-\mu) \leq \sigma(0) = 0.5$ and $\ell = -\log(\sigma(s-\mu)) \geq -\log(0.5) = \log 2$. We can find that the loss is substantial, indicating that the model needs to increase the separation between preferred and non-preferred responses. As $s - \mu \to -\infty$, $\sigma(s-\mu) \to 0, \quad \ell \to \infty$.

2. If $s > \mu$ (Easy Sample, margin satisfied), we will have $\sigma(s-\mu) > 0.5$ and $\ell = -\log(\sigma(s-\mu)) < \log 2$. We observe that as $s - \mu \to \infty$, $\sigma(s-\mu) \to 1, \quad \ell \to 0$. This shows that the loss function penalizes more heavily when the score difference s does not exceed the margin μ.

Impact on Gradient. We are also interested in the gradient of the loss ℓ with respect to the score difference s, which dictates how the model scores are updated. Let $X = s - \mu$, then we can obtain the following derivation:

$$\frac{\partial \ell}{\partial X} = \frac{d}{dX}\left(-\log(\sigma(X))\right) = -\frac{1}{\sigma(X)} \cdot \sigma'(X) \tag{11}$$

Since $\sigma'(X) = \sigma(X)(1-\sigma(X)), \quad$ then $\frac{\partial \ell}{\partial X} = -(1-\sigma(X)) = \sigma(X) - 1$. By chain rule:

$$\frac{\partial \ell}{\partial s} = \frac{\partial \ell}{\partial X} \cdot \frac{\partial X}{\partial s} = \sigma(s-\mu) - 1 \tag{12}$$

In gradient descent, model parameters are updated in the direction of the negative gradient. Therefore, the effective update signal for increasing s is proportional to:

$$-\frac{\partial \ell}{\partial s} = 1 - \sigma(s-\mu) \tag{13}$$

1. If $s \leq \mu$ (Difficult/Marginal Sample), we will have $\sigma(s-\mu) < 0.5, \quad 1-\sigma(s-\mu) > 0.5$. As $s - \mu \to -\infty$, $\sigma(s-\mu) \to 0, \quad 1-\sigma(s-\mu) \to 1$, we obtain a strong gradient signal, encouraging the model to increase s for hard pairs.

2. If $s > \mu$ (Easy Sample), we will have $\sigma(s-\mu) > 0.5, \quad 1-\sigma(s-\mu) < 0.5$. As $s - \mu \to \infty$, $1-\sigma(s-\mu) \to 0$, we observe that the update signal vanishes, reflecting that learning pressure reduces once margin is satisfied.

This analysis confirms that the model primarily updates its parameters to increase the score difference s when $s \leq \mu$. Once $s > \mu$, the incentive to further increase s diminishes. Therefore, a larger μ maintains learning pressure across a wider range of s values, promoting more substantial separation between preferred and non-preferred responses.

B RM Training Details

Without specific statement, we set $\gamma = 0.5$ in equation 6 and $\beta = 0.1$ in OT. During reward model training, we adopt the LoRA and train all the reward models for 2 epochs using a learning rage of 4×10^{-5}. The inputs are truncated by 1024 tokens and more detailed hyper-parameters can be found in Table 6.

C PPO Training and Evaluation Details

For our PPO experiment, we fine-tune two distinct models using 20,000 samples from the alpaca-gpt4-data-en dataset (Baolin Peng, 2023). The first model, Llama3.1-8B-Instruct[3], has undergone post-training

[3] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

Table 6: Implementation details.

Quantization	bf16
LoRA r	32
LoRA α	64
LoRA dropout	0.05
Optimizer	Adamw_hf
Global Batch Size	128
Learning Rate	2×10^{-6}
Learning Rate Scheduler	cosine
RM-GPUs	1×4 GPU Cards
RLHF-GPUs	1×8 GPU Cards
Warmup Ratio	0.03

that includes both DPO and RLHF. The second, OpenRLHF-Llama3-8B-SFT[4], is an instruction-following version built upon Llama3-8B-Base, without RLHF post-training stage. We conduct the PPO training using the ms-swift framework[5] with its default training configuration. All benchmark evaluations are subsequently performed using the ms-evalscope framework[6]. Our evaluation protocol utlize few-shot settings for GSM8K (4-shot) (Cobbe et al., 2021a), Race (3-shot) (Lai et al., 2017), and TriviaQA (5-shot) (Joshi et al., 2017), while all other benchmarks (i.e., Hellaswag (Zellers et al., 2019), IFeval (Zhou et al., 2023), MMLU (Hendrycks et al., 2021), ProcessBench (Zheng et al., 2025), BBH (Suzgun et al., 2022b), and Humaneval (Chen et al., 2021)) are assessed in a zero-shot setting. We report accuracy as the primary metric for all tasks, with the exception of Humaneval, for which we report the Pass@1 score.

[4] https://huggingface.co/OpenRLHF/Llama-3-8b-sft-mixture
[5] https://github.com/modelscope/ms-swift
[6] https://github.com/modelscope/evalscope