
On the Robustness of Reward Models for Language Model Alignment

Jiwoo Hong¹ Noah Lee¹ Eunki Kim¹ Guijin Son² Woojin Chung¹
Aman Gupta³ Shao Tang³ James Thorne¹

Abstract

The Bradley-Terry (BT) model is widely practiced in reward modeling for reinforcement learning with human feedback (RLHF). Despite its effectiveness, reward models (RMs) trained with BT model loss are prone to *over-optimization*, losing generalizability to unseen input distributions. In this paper, we study the cause of over-optimization in RM training and its downstream effects on the RLHF procedure, accentuating the importance of distributional robustness of RMs in unseen data. First, we show that the excessive dispersion of hidden state norms is the main source of over-optimization. Then, we propose batch-wise sum-to-zero regularization (**BSR**) to enforce zero-centered reward sum per batch, constraining the rewards with extreme magnitudes. We assess the impact of BSR in improving robustness in RMs through four scenarios of over-optimization, where BSR consistently manifests better robustness. Subsequently, we compare the plain BT model and BSR on RLHF training and empirically show that robust RMs better align the policy to the gold preference model. Finally, we apply BSR to high-quality data and models, which surpasses state-of-the-art RMs in the 8B scale by adding more than 5% in complex preference prediction tasks. By conducting RLOO training with 8B RMs, AlpacaEval 2.0 reduces generation length by 40% while adding a 7% increase in win rate, further highlighting that robustness in RMs induces robustness in RLHF training. We release the code, data, and models: <https://github.com/LinkedIn-XFACT/RM-Robustness>.

1. Introduction

Reward models (RMs) are crucial components in reinforcement learning with human feedback (RLHF), being *proxies* for human preference for aligning large language models (LLMs) (Christiano et al., 2017; Ziegler et al., 2020; Casper et al., 2023; Wang et al., 2024a). RMs map text sequences to a scalar score, and are incorporated into the RLHF pipeline by directly using the scores (Ziegler et al., 2020; Ahmadian et al., 2024), or labeling pairwise preferences (Rafailov et al., 2023; Hong et al., 2024; Meng et al., 2024).

The Bradley-Terry (BT) model (Bradley & Terry, 1952) formulation is widely adopted to train RMs using a tuple of a prompt and two corresponding responses with a pre-annotated preference relation (e.g., chosen and rejected), maximizing the margin of rewards between the responses. Recent works proposed *additive approaches* to the BT model to further adjust it to the neural language model context (Yuan et al., 2024; Coste et al., 2024; Yang et al., 2024b). However, these do not adjust the learning objective of the BT model at a fundamental level.

Reward model over-optimization, where RMs overfit to the train set and eventually losing the alignment capability to the true preference distribution, is a typical limitation of neural RMs (Gao et al., 2023; Coste et al., 2024). Efforts to curate high-quality pairwise preference datasets has contributed to enhancing RMs (Yuan et al., 2024; Liu et al., 2024a; Cui et al., 2025), assessed through pairwise preference benchmarks (Kim et al., 2024; Lambert et al., 2025; Liu et al., 2025b). However, using LLMs either as preference data generators or annotators often induces inherent biases like verbosity bias and self-enhancement biases (Saito et al., 2023; Zhang et al., 2024b; Chen et al., 2024a) and obstructs understanding of underlying RM over-optimization.

In this paper, we show that reward modeling with the BT model induces *excessive hidden state norm dispersion*, leading to the over-optimization problem. We propose batch-wise sum-to-zero regularization (**BSR**) as a straightforward solution to control the dispersion by penalizing abnormal reward outliers, eventually reducing the dispersion in hidden state norms. We categorize four generalization scenarios for RMs with respect to prompt and response space, allowing

¹KAIST AI ²OneLineAI ³LinkedIn Corporation. Correspondence to: Jiwoo Hong <jiwoo_hong@kaist.ac.kr>.

fine-grained analysis of over-optimization and robustness. Based on these scenarios, we propose a regularized reward modeling objective $\mathcal{L}_{\text{BT-BSR}}$, excelling baseline methods in all generalization scenarios and improving accuracy in complex tasks over 5% in RM-Bench compared to simple BT model. Further analysis of the propagation of robustness in RMs to RLHF training shows **BSR** yields a 40% decrease in generation length with 7% gain in AlpacaEval 2.0.

2. Background

2.1. Preliminaries

Reward modeling with language models Reward models (RMs) for language model alignment use the Bradley-Terry (BT) model (Bradley & Terry, 1952) for human preference between chosen and rejected responses y_w and y_l given a prompt x as:

$$P(y_w \succ y_l | x) = \frac{e^{r(x, y_w)}}{e^{r(x, y_w)} + e^{r(x, y_l)}}. \quad (1)$$

Typically, they are implemented as classifiers with projection head W_p (Ziegler et al., 2020; Ouyang et al., 2022):

$$r(x, y) = W_p^T \cdot h(x, y), \quad W_p \in \mathbb{R}^{H \times 1}, \quad (2)$$

$h(x, y)$ refers to the last hidden state from the backbone language model with the hidden dimension of H , given the prompt and response pair (x, y) . In practice, both encoder-only (Jiang et al., 2023b) and decoder-only language models (Yuan et al., 2024; Liu et al., 2024a) are used as a backbone model that returns $h(x, y)$. And W_p is the projection head which the weights are randomly initialized by $\mathcal{N}(0, (H + 1)^{-1})$ (Stiennon et al., 2020; Huang et al., 2024).

RMs are fine-tuned to maximize the margin between the chosen and rejected responses' scores (Christiano et al., 2017; Stiennon et al., 2020):

$$\mathcal{L}_{\text{BT}} = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \frac{e^{r(x, y_w)}}{e^{r(x, y_w)} + e^{r(x, y_l)}} \right], \quad (3)$$

where \mathcal{D} is the set of triplets comprising chosen and rejected responses y_w and y_l given the fixed prompt x , and $r(x, y) \in \mathbb{R}$ is a score assigned to (x, y) .

Preference modeling with BT model BT model is defined for *directly comparable* components (Bradley & Terry, 1952; Davidson, 1970; Huang et al., 2004), leaving the responses $y_{i,j} \sim \mu(\cdot | x_i)$ given the fixed prompt x_i to be comparable. For each prompt x_i , there exists a BT model with prompt-specific parameters ϕ_i that defines the preference:

$$P(y_{i,1} \succ y_{i,2} | x_i; \phi_i) = \frac{e^{s_{\phi_i}(x_i, y_{i,1})}}{e^{s_{\phi_i}(x_i, y_{i,1})} + e^{s_{\phi_i}(x_i, y_{i,2})}}, \quad (4)$$

where $s_{\phi_i}(x_i, \cdot)$ represents the scoring function specific to prompt x_i . However, the reward model r_θ parameterized by the language model unifies these prompt-specific BT models by learning a single parameterization θ that maps the entire space of prompts and responses $\mathcal{X} \times \mathcal{Y}$ to preference scores.

Reward model parameterization Let $\mathcal{M} = M_1, \dots, M_K$ be a set of language models where each M_k defines a conditional distribution over responses given a prompt. For any prompt x in the prompt set $\mathcal{X} = \{x_i\}_{i=1}^N$, each model M_k induces a prompt-specific response space $\mathcal{Y}_k(x) = \text{supp}(M_k(\cdot | x))$:

$$\mathcal{Y}(x) = \bigcup_{k=1}^K \mathcal{Y}_k(x), \quad (5)$$

where $\mathcal{Y}(x)$ represents all possible responses for prompt x across all models. Thus, r_θ can be represented as:

$$r_\theta : \mathcal{X} \times \mathcal{Y}(x) \rightarrow \mathbb{R}, \quad (6)$$

$$\text{where } s_{\phi_i}(x_i, \cdot) = r_\theta(x_i, \cdot) \quad \forall x_i \in \mathcal{X}. \quad (7)$$

This shared parameterization enables learning preference patterns that generalize across the prompt space while maintaining the BT structure within each prompt context.

2.2. Problem setup: robustness of reward models

Motivated by how RMs parameterize multiple prompt-level BT models into a single parameter θ , we expand the issue of *reward model over-optimization* (Gao et al., 2023) by categorizing generalization scenarios based on prompt disjointness and response disjointness. By doing so, **we study how each generalization scenario of RM is affected by over-optimization and how to improve the robustness.**

We follow how Gao et al. (2023) sets the synthetic gold RM instead of human annotations for controlled assessment. Let there exist a true preference model $r^* : \mathcal{X} \times \mathcal{Y}(x) \rightarrow \mathbb{R}$. Let $\mathcal{M}_{\text{train}} \subset \mathcal{M}$ and $\mathcal{M}_{\text{valid}} \subset \mathcal{M}$ be the sets of models used for generating responses in the training and validation sets respectively. The corresponding response spaces are:

$$\mathcal{Y}_{\text{train}}(x) = \bigcup_{M_k \in \mathcal{M}_{\text{train}}} \mathcal{Y}_k(x) \quad (8)$$

$$\mathcal{Y}_{\text{valid}}(x) = \bigcup_{M_k \in \mathcal{M}_{\text{valid}}} \mathcal{Y}_k(x). \quad (9)$$

Thus, the train set $\mathcal{D}_{\text{train}}$ for training r_θ can be defined as:

$$\mathcal{D}_{\text{train}} = \{(x, y_w, y_l) | (y_w, y_l) \sim \mathcal{Y}_{\text{train}}(x), x \in \mathcal{X}_{\text{train}}\}. \quad (10)$$

First, *prompt disjointness* is defined as two prompt sets $\mathcal{X}_{\text{train}}$ and $\mathcal{X}_{\text{valid}}$ being disjoint. And *response disjointness* is defined as two response spaces $\mathcal{Y}_{\text{train}}$ and $\mathcal{Y}_{\text{valid}}$ having disjoint response model sets $\mathcal{M}_{\text{train}}$ and $\mathcal{M}_{\text{valid}}$. From this context, we categorize RM generalization scenarios into:

1. **In-domain (\mathcal{D}_{ID}):** \mathcal{D}_{ID} shares the same prompts with $\mathcal{D}_{\text{train}}$ and responses are sampled from the same response model set $\mathcal{M}_{\text{train}}$:

$$\{(x, y_w, y_l) | (y_w, y_l) \sim \mathcal{Y}_{\text{train}}(x), x \in \mathcal{X}_{\text{train}}\}. \quad (11)$$

2. **Prompt-disjoint ($\mathcal{D}_{\sim \text{Prompt}}$):** r_θ being generalizable to the unseen prompt set $\mathcal{X}_{\text{valid}}$ but with the same response model set $\mathcal{M}_{\text{train}}$:

$$\{(x, y_w, y_l) | (y_w, y_l) \sim \mathcal{Y}_{\text{train}}(x), x \in \mathcal{X}_{\text{valid}}\}. \quad (12)$$

3. **Response-disjoint ($\mathcal{D}_{\sim \text{Response}}$):** r_θ being generalizable to the seen prompt set $\mathcal{X}_{\text{train}}$ but with the unseen response model set $\mathcal{M}_{\text{valid}}$:

$$\{(x, y_w, y_l) | (y_w, y_l) \sim \mathcal{Y}_{\text{valid}}(x), x \in \mathcal{X}_{\text{train}}\}. \quad (13)$$

4. **Mutual-disjoint ($\mathcal{D}_{\sim \text{Mutual}}$):** r_θ being generalizable to both unseen prompts $\mathcal{X}_{\text{valid}}$ and response models $\mathcal{M}_{\text{valid}}$:

$$\{(x, y_w, y_l) | (y_w, y_l) \sim \mathcal{Y}_{\text{valid}}(x), x \in \mathcal{X}_{\text{valid}}\}. \quad (14)$$

From this context, we define RM over-optimization as RM’s accuracy on $\mathcal{D}_{\text{train}}$ and \mathcal{D}_{ID} increasing, while its performance on $\mathcal{D}_{\sim \text{Prompt}}$, $\mathcal{D}_{\sim \text{Response}}$, and $\mathcal{D}_{\sim \text{Mutual}}$ stagnate or degrades. Intuitively, it is equivalent to r_θ losing its alignment with r^* in the general cases after training with limited samples.

2.3. Post-analysis: Reinforcement learning with human feedback

We then analyze the propagation of over-optimization in the downstream RLHF applications. By fine-tuning the language model π with reinforcement learning (RL) algorithms to maximize the reward with respect to r_θ as human preference proxies, the policy π is trained to maximize the average reward for its response $y \sim \pi(\cdot|x)$ given the prompt $x \sim \mathcal{D}$:

$$\max_{\pi} \mathbb{E}_{y \sim \pi(\cdot|x), x \sim \mathcal{D}} [r_\theta(x, y)] - \beta \mathbb{D}_{\text{KL}}(\pi(y|x) | \pi_{\text{ref}}(y|x)). \quad (15)$$

To understand the impact of over-optimization in RLHF, we assess the gold reward scores of interim π during the training. Doing so, **we verify if maximizing $r(x, y)$ is aligned with maximizing $r^*(x, y)$.**

3. Experiments

We set ArmoRM (Wang et al., 2024b) as the gold preference model r^* , a setup similar to that of Gao et al. (2023). We selected ArmoRM as it utilizes a wide range of representative synthetic preference data and is reported to be robust to various biases (Wang et al., 2024b; Meng et al., 2024).

3.1. Part 1: Reward model over-optimization

Models We use two different model families, Llama-3 (Dubey et al., 2024) and Qwen2.5 (Yang et al., 2024a), with

varying sizes. We select Llama-3.2-1B, 3B, and Llama-3.1-8B base models from the Llama-3, along with Qwen2.5-1.5B, 3B, and 7B base models from the Qwen2.5. We use UltraChat (Ding et al., 2023, UC) to conduct supervised fine-tuning (SFT) for every model. Detailed SFT and reward modeling training configurations are in Appendix A.

Datasets We adopt UltraFeedback (Cui et al., 2025, UF), which harnesses 17 different models with varying model families to sample four responses per prompt, to set one train set and four validation sets as described in Section 2.2.

We set the original 17 models as $\mathcal{M}_{\text{train}}$ and select four disjoint models as $\mathcal{M}_{\text{valid}}$: Gemma-2-2B-It (Team et al., 2024), Olmo2-7B-Instruct (OLMo et al., 2025), SmolLM2-1.7B-Instruct (Allal et al., 2025), and Mistral-Instruct-v0.2 (Jiang et al., 2023a). We excluded the Llama and Qwen families to avoid contamination with the training models and $\mathcal{M}_{\text{train}}$. After generating four more responses per prompt with them, we prepare a train set $\mathcal{D}_{\text{train}}$ with 51,200 rows and four validation sets: \mathcal{D}_{ID} , $\mathcal{D}_{\sim \text{Prompt}}$, $\mathcal{D}_{\sim \text{Response}}$, and $\mathcal{D}_{\sim \text{Mutual}}$. We report detailed preprocessing procedures in Appendix B.

3.2. Part 2: Propagation of RM robustness in RLHF

We simulate the impact of robustness in RM during reinforcement learning with human feedback (RLHF) with RLOO (Ahmadian et al., 2024). We test if the trained model with each RM can align with the gold preference model r^* .

Models and dataset We set the Qwen2.5-1.5B base model with supervised fine-tuning (SFT) applied using UC as an initial policy. We employ Qwen2.5-3B based RMs trained with \mathcal{L}_{BT} and $\mathcal{L}_{\text{BT-BSR}}$ from Section 3.1, as they demonstrated the highest overall performance in Figure 5 for both methods. We use $\mathcal{X}_{\text{valid}}$ as the seed prompts for experiments. We report the additional training configurations in Table 4.

3.3. Part 3: Real-world impact of robustness in RM

Finally, we extend our experiments in Sections 3.1 and 3.2 to the 8B model and high-quality synthetic preference data to demonstrate the scalability and effectiveness of the proposed method.

Models and datasets We use TULU3 SFT mixture (Lambert et al., 2024) to conduct SFT on Qwen2.5-1.5B. Then, we employ Llama-3.1-8B based RMs trained with \mathcal{L}_{BT} and $\mathcal{L}_{\text{BT-BSR}}$ on Skywork-Reward-Preference-80K-v0.2¹, high-quality synthetic preference dataset (Liu et al., 2024a; 2025b). We use the same $\mathcal{X}_{\text{valid}}$ as the seed prompts for RLOO. We report the additional training configurations in Table 4.

¹<https://huggingface.co/datasets/Skywork/Skywork-Reward-Preference-80K-v0.2>

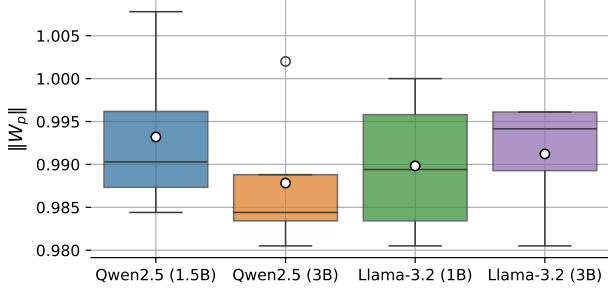


Figure 1. $\|W_p\|$ distribution after reward modeling for four seeds each. $\|W_p\|$ generally stays around one after the training.

4. Robustness and Over-optimization in RMs

In this section, we propose excessive dispersion in the hidden states of the reward model (RM) as a cause of over-optimization. We support our point by analyzing the training dynamics of \mathcal{L}_{BT} in Section 4.1. Then, we propose *batch-wise sum-to-zero regularization* (BSR) as a method to mitigate such dispersion, enhancing the robustness of RMs.

4.1. Hypothesis

Given the prompt and response pair (x, y) , the final score of RM r_θ as a dot product between the projection head W_p and the hidden state $h(x, y)$ can be decomposed into:

$$r_\theta(x, y) = \|W_p\| \cdot \|h(x, y)\| \cdot \cos \psi, \quad (16)$$

where $\cos \psi$ represents the cosine similarity between the two vectors $W_p \in \mathbb{R}^{H \times 1}$ and $h(x, y) \in \mathbb{R}^{H \times 1}$. Typically, the norm of two vectors $\|W_p\| \cdot \|h(x, y)\|$ largely contributes to maximize the softmax value and cause over-confidence issue (Wei et al., 2022), especially in the context of large language models (LLMs) that have large hidden sizes (Yang et al., 2024a; Dubey et al., 2024). Inspired by Wei et al. (2022), we hypothesize that excessive growth of hidden state norm dispersion is a major cause of RM over-optimization.

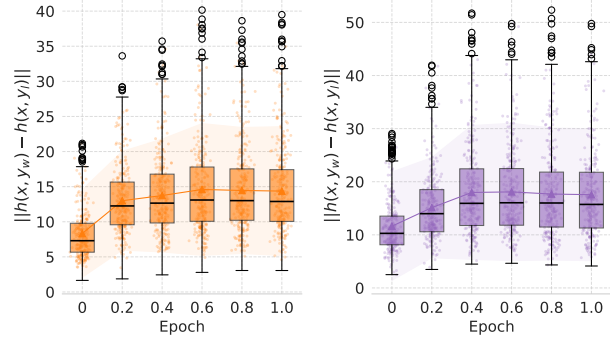
Chosen and rejected responses share projection head

Within three components in Equation (16), we first analyze the contribution of W_p to over-optimization. Let Δr the reward margin $r(x, y_w) - r(x, y_l)$ for the prompt x and chosen and rejected responses y_w and y_l :

$$\Delta r = W_p^T (h(x, y_w) - h(x, y_l)). \quad (17)$$

As Equation (3) is equivalent to:

$$\mathcal{L}_{BT} = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} [\log \sigma(\Delta r)], \quad (18)$$



(a) Qwen2.5 (3B)

(b) Llama-3.2 (3B)

Figure 2. Growth of $\|h(x, y_w) - h(x, y_l)\|$ throughout reward modeling with \mathcal{L}_{BT} . The variance of the hidden state difference grows incrementally with the right-skewed distribution. The width of the colored area indicates the standard deviation.

the gradient updates for W_p can be written as:

$$\frac{\partial \mathcal{L}_{BT}}{\partial W_p} = -(1 - \sigma(\Delta r)) \cdot \frac{\partial \Delta r}{\partial W_p} \quad (19)$$

$$\frac{\partial \Delta r}{\partial W_p} = h(x, y_w) - h(x, y_l), \quad (20)$$

where $\sigma(x) = (1 + \exp(-x))^{-1}$ is the sigmoid function. As $\sigma(-x) = 1 - \sigma(x)$,

$$\frac{\partial \mathcal{L}_{BT}}{\partial W_p} = -\sigma(-\Delta r) \cdot (h(x, y_w) - h(x, y_l)). \quad (21)$$

Then, the gradient norm for W_p is:

$$\left\| \frac{\partial \mathcal{L}_{BT}}{\partial W_p} \right\| = \sigma(-\Delta r) \cdot \|h(x, y_w) - h(x, y_l)\|. \quad (22)$$

Having $\sigma(-\Delta r)$ and $\|h(x, y_w) - h(x, y_l)\|$ as main components, the gradient norm of W_p saturates as \mathcal{L}_{BT} is minimized by maximizing Δr . Despite $\Delta r \simeq 0$ in the initial phase of training, $\|h(x, y_w) - h(x, y_l)\|$ is minimal at the beginning as the output hidden states of large language models (LLMs) (*i.e.*, backbone LM of RMs) have low effective rank, being concentrated to certain regions (Biś et al., 2021; Wei et al., 2024).

Thus, $\|W_p\|$ is expected to have marginal difference with its initial value, which is $\mathbb{E}[\|W_p\|] = 1$ by $\mathcal{N}(0, 1/(H + 1))$ initialization (Stiennon et al., 2020). We support this through Figure 1, average $\|W_p\|$ across four seeds for each model after training with \mathcal{L}_{BT} actually stays near 1.

Norm variance inflates in hidden states with \mathcal{L}_{BT} We continue by analyzing the impact of $\|h(x, y)\|$ in over-optimization. $\min_\theta \mathcal{L}_{BT}$ can be achieved through maximizing Δr , the inner term of sigmoid function:

$$\Delta r_\theta = \|W_p\| \cdot \|h_\theta(x, y_w) - h_\theta(x, y_l)\| \cdot \cos \psi_\Delta, \quad (23)$$

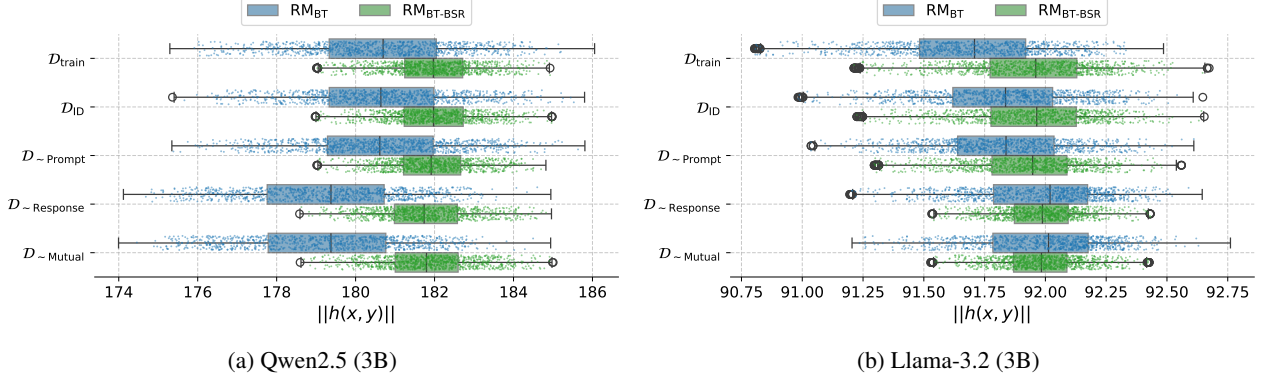


Figure 3. Hidden state norm dispersion comparison between RM_{BT} and $\text{RM}_{\text{BT-BSR}}$. Batch-wise sum-to-zero regularization (**BSR**) alleviates hidden state norm dispersion and demonstrates a consistent range of norms across different generalization scenarios in Section 2.2. Reducing the variability of hidden state norms, \mathcal{L}_{BSR} improves the robustness of RMs for unseen data.

when $\cos \psi_{\Delta}$ is the cosine similarity between W_p and $h_{\theta}(x, y_w) - h_{\theta}(x, y_l)$. Having $\|W_p\|$ stays near its initial value of one, \mathcal{L}_{BT} encourages θ to increase $\|h_{\theta}(x, y_w) - h_{\theta}(x, y_l)\|$ and $\cos \psi_{\Delta}$.

Over-confidence and over-optimization in RMs In a multi-class classification task, Wei et al. (2022) points out that excessive growth of logit magnitude in classifiers causes overfitting in the training set: *i.e.*, over-confidence issue. This is closely connected to the over-optimization problem as the reward models are simply classifiers optimized with two-class classification objective \mathcal{L}_{BT} . Viewing $r_{\theta}(x, y)$ (16) as a logit in a classifier, our analysis narrows the cause of growth in $\|r_{\theta}(x, y)\|$ to the growth of $\|h_{\theta}(x, y_w) - h_{\theta}(x, y_l)\|$ as $\|W_p\|$ remains near one.

In Figure 2, we track the growth of $\|h_{\theta}(x, y_w) - h_{\theta}(x, y_l)\|$ while fine-tuning RM with \mathcal{L}_{BT} . Compared to that of hidden states from the SFT model (Epoch 0), we observe a consistent increase in the average norm and its variance, especially strengthening the right-skewness. This aligns with how the over-confidence issue happens due to growing logit magnitude in typical classification tasks (Wei et al., 2022).

4.2. Method: Batch sum-to-zero regularization (BSR)

From this context, we propose regularizing the reward sum of batch to zero to control the inflation in hidden state norm variance, namely *batch sum-to-zero regularization (BSR)*:

$$\mathcal{L}_{\text{BT-BSR}} = \mathcal{L}_{\text{BT}} + \lambda \cdot \mathcal{L}_{\text{BSR}} \quad (24)$$

$$\mathcal{L}_{\text{BSR}} = \left(\frac{1}{2|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \sum_{j \in \{w, l\}} r(x_i, y_{i,j}) \right)^2, \quad (25)$$

where λ is the weight hyperparameter and \mathcal{B} refers to the batch of (x, y_w, y_l) . \mathcal{L}_{BSR} penalizes $r(x, y)$ from being

skewed by enforcing the reward sum to zero, constraining the norm dispersion and outliers in Figure 2.

In detail, the gradient for $h(x, y_w)$ and $h(x, y_l)$ are symmetric, having same magnitude but opposite directions:

$$\frac{\partial \mathcal{L}_{\text{BT}}}{\partial h(x, y_w)} = -\sigma(-\Delta r) W_p \quad (26)$$

$$\frac{\partial \mathcal{L}_{\text{BT}}}{\partial h(x, y_l)} = \sigma(-\Delta r) W_p. \quad (27)$$

By \mathcal{L}_{BT} , $\|h(x, y_w) - h(x, y_l)\|$ incrementally increases as Figure 2. Meantime, \mathcal{L}_{BSR} penalizes outliers with large magnitude toward either positive or negative directions, shown through its gradient:

$$\frac{\partial \mathcal{L}_{\text{BSR}}}{\partial h(x_i, y_{i,j})} = \left(\frac{1}{|\mathcal{B}|} r(x_i, y_{i,j}) \right) \cdot W_p, \quad (28)$$

for a prompt and response pair $(x_i, y_{i,j})$ with $i \in \{1, \dots, |\mathcal{B}|\}$ and $j \in \{w, l\}$, preventing excessive growth as the gradient is proportional to $r(x, y)$. This is a straightforward solution for norm variance inflation in Figure 2, as $r(x, y)$ and its dispersion are largely influenced by $\|h(x, y)\|$ as discussed in Section 4.1, especially as $r_{\theta}(x, y_w)$ and $r_{\theta}(x, y_l)$ are positive and negative values.

We support this point through Figure 3, RM trained with $\mathcal{L}_{\text{BT-BSR}}$ having significantly lower dispersion in $\|h(x, y)\|$. Comparing the variability of $\|h(x, y)\|$ on four different generalization scenarios in Section 2.2, reward model trained with $\mathcal{L}_{\text{BT-BSR}}$ ($\text{RM}_{\text{BT-BSR}}$) is robust to unseen prompt and responses, while the norm range differs with reward model trained with \mathcal{L}_{BT} (RM_{BT}).

4.3. Baselines

Based on Section 4.1, we adopt three algorithmic mitigations as baseline methods to fulfill Equation (23).

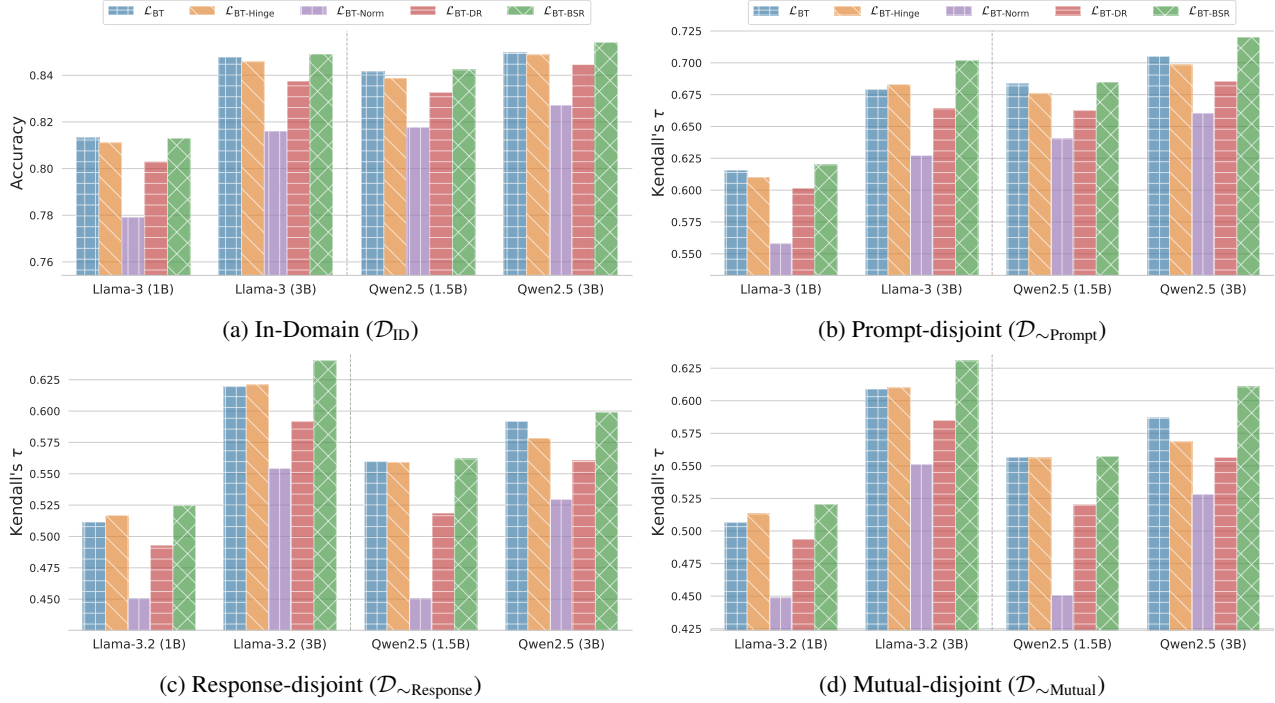


Figure 4. Assessing the robustness of each reward modeling objective on four generalization scenarios in Section 2.2. By applying \mathcal{L}_{BT-BSR} , downstream RMs are more robust to unseen prompt sets or response generation model sets by being more aligned to the gold preference model r^* , measured through accuracy and Kendall’s τ against the preference annotations of r^* . Full results are reported in Table 5.

First, we can explicitly normalize rewards with their norm. Logit normalization (Wei et al., 2022) can be applied to \mathcal{L}_{BT} as it isolates classification objective to the directions of logits, $\cos \psi$, without being affected by $\|h(x, y)\|$ by normalizing with the norm of rewards:

$$\mathcal{L}_{BT-Norm} = -\log \sigma \left(\frac{r(x, y_w) - r(x, y_l)}{\sqrt{r(x, y_w)^2 + r(x, y_l)^2}} \right). \quad (29)$$

Second, we can set the hardbound of Δr to explicitly prevent the divergence in Equation (23). We add the hinge loss (Cortes, 1995) that provides hardbound m :

$$\mathcal{L}_{BT-Hinge} = \max(0, m - (r(x, y_w) - r(x, y_l))). \quad (30)$$

Third, we test the extreme case of boosting the margin in Equation (23), which is expected to reinforce the over-confidence issue. We find the loss function in Yuan et al. (2024) additionally boosting the margin with separate logistic losses for chosen and rejected responses, respectively:

$$\mathcal{L}_{BT-DR} = \mathcal{L}_{BT} - \log \sigma(r_\theta(x, y_w)) - \log \sigma(-r_\theta(x, y_l)). \quad (31)$$

Based on our analysis in Section 4.1, we hypothesize that \mathcal{L}_{BT-DR} will underperform compared to \mathcal{L}_{BT} in all generalization scenarios by booting the hidden state norm dispersion.

5. Results and Analysis

Through three steps, we analyze the significance of robustness in reward models (RMs) in reinforcement learning with human feedback (RLHF). Beginning with how each method affects robustness in RMs through over-optimization assessment in Section 5.1, we study how the robustness in RMs as proxies can boost the alignment toward the true preference in Section 5.2. Finally, we expand our scope to state-of-the-art data and models in Section 5.3.

5.1. Part 1: Reward model over-optimization

In Figure 4, we assess the robustness of reward models (RMs) trained with each method in Section 4.3 on four generalization scenarios discussed in Section 2.2. The full results are reported in Appendix C.

RMs are prone to unseen response styles Comparing Figures 4b and 4d, most of the RMs were prone to unseen response styles. While both $\mathcal{D}_{\sim Prompt}$ and $\mathcal{D}_{\sim Mutual}$ shares the same prompt set *unseen* during the training, adding an unseen response model set $\mathcal{M}_{\text{valid}}$ triggers more than 10% loss in Kendall’s τ values. For instance, RM_{BT} experienced 16.4% decrease, from τ of 0.705 to 0.587. This implies that harnessing diverse LLMs in synthetic preference datasets is crucial for the general use of RMs.

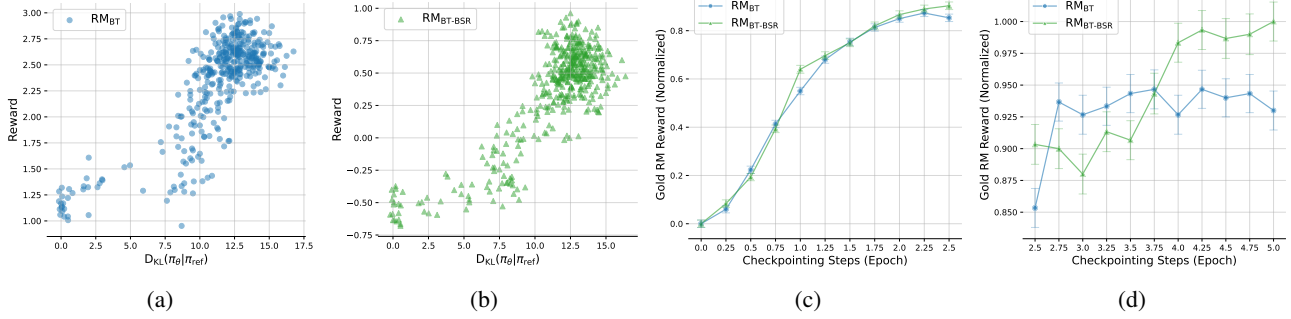


Figure 5. Reward to KL divergence plots (Figures 5a and 5b) and alignment to gold reward model r^* (Figures 5c and 5d) while fine-tuning Qwen2.5-1.5B SFT model as π_θ with RLOO using RM_{BT} and RM_{BT-BSR} , respectively. We measure the alignment with the gold preference model r^* to verify if reward maximization with each RM as a proxy is leading to maximizing r^* scores in Figures 5c and 5d. Using RM_{BT-BSR} consistently improved the reward assessed by r^* , while RM_{BT} stagnated in the last half of the training.

BSR enhances the robustness of RMs In Figure 4, RM_{BT-BSR} best aligns to r^* across all generalization scenarios. Meantime, the accuracy of RM_{BT-DR} , which is expected to have even stronger norm dispersion, is consistently lower than RM_{BT} , evidently supporting our hypothesis. With Figure 3, we inspect that $\|h(x, y)\|$ remaining stable with unseen data is a crucial condition for robustness in RMs.

Also, $RM_{BT-Hinge}$ occasionally exceeded RM_{BT} in Figures 4c and 4d. Recall that $\mathcal{L}_{BT-Hinge}$ sets hardbound to restrict the divergence in Equation (23); such a result supports that the growth of $\|h(x, y)\|$ is a major cause of over-optimization as discussed. However, explicit logit normalization with $\mathcal{L}_{BT-Norm}$ underperformed in every scenario. This implies that $\|h(x, y)\|$ shouldn’t be excluded in reward prediction, highlighting the necessity of soft regularization.

Finally, we observe that the performance gap between RM_{BT} and RM_{BT-BSR} is amplified as the model size increases in all four plots, comparing within the same model families. We further study this in Section 5.3 with Llama-3.1-8B.

5.2. Part 2: Propagation of RM robustness in RLHF

We study how robust RMs improve RLHF tasks by using RM_{BT} and RM_{BT-BSR} for RLOO training in Figure 5. We refer to π_{BT} and π_{BT-BSR} to be the two RLOO-trained policies with Qwen2.5-3B based RM_{BT} and RM_{BT-BSR} .

Robustness in RMs lead to better alignment of π with r^* In Figures 5c and 5d, we observe π_{BT-BSR} being better aligned to the gold preference model r^* (i.e., ArmoRM). While π_{BT} and π_{BT-BSR} both incrementally maximized the reward in terms of RM_{BT} and RM_{BT-BSR} , Figure 5d depicts that the r^* scores *stagnate* in the last half for RLOO training for π_{BT} . This could imply RM_{BT} providing an imperfect reward signal in the high-reward region, where $\|h(x, y)\|$ tends to be relatively large. This aligns with the observation in Figure 3 where the average $\|h(x, y)\|$ of RM_{BT} was

Table 1. Effective rank analysis for RM_{BT} and RM_{BT-BSR} by collecting the hidden states of each model. While RM_{BT} structures the train data into a lower rank, the effective rank significantly increases on the unseen evaluation dataset from RM-Bench.

	$erank_{train}$	$erank_{eval}$	$\Delta (\downarrow)$
RM_{BT}	23.34	33.76	+10.42
RM_{BT-BSR}	33.45	33.59	+0.14

sensitive to the unseen data.

BSR and stable reward maximization Through Figures 5a and 5b, we compare the stability in reward maximization by tracking the divergence of π_θ . With RM_{BT} , initial reward maximization required a leap in KL divergence, as shown in the 0.0 to 7.5 range in Figure 5a. On the other hand, RM_{BT-BSR} induced continuous exponential growth in reward while π_{BT-BSR} diverged. Along with steady increase in r^* score, this implies that RM_{BT-BSR} yield stable training.

5.3. Part 3: Real-world impact of robustness in RM

We apply our method and insights to the state-of-the-art preference dataset and the most widely practiced base model, Skywork-Preferences-v0.2 (SKP) and Llama-3.1-8B, where we do not have access to the true preference model. For RM_{BT} , we use the official checkpoint of Liu et al. (2024a)².

Parametric robustness analysis for RM_{BT} and RM_{BT-BSR} Along with RM-bench, we employ the effective rank (Roy & Vetterli, 2007; Wei et al., 2024, $erank$) as a parametric analysis of robustness. The large discrepancy between the $erank$ of the train and evaluation sets indicates that the model fails to capture information that does not exist in the train

²<https://huggingface.co/Skywork/Skywork-Reward-Llama-3.1-8B-v0.2>

Table 2. RM-Bench comparison between RMs trained with RM_{BT} and $\text{BT}_{\text{BT-BSR}}$ with varying $\lambda \in \{10^{-2}, 10^{-3}, 10^{-4}\}$. By setting $\lambda = 0$, $\text{BT}_{\text{BT-BSR}}$ is equivalent to RM_{BT} . As λ gets larger, accuracy in hard tasks with subtle differences increases, consistently exceeding RM_{BT} .

RM-Bench	Chat	Math	Code	Safety	Hard Acc	Normal Acc	Easy Acc	Overall
$\text{RM}_{\text{BT}} (\lambda = 0)$	69.2	62.1	53.4	95.9	47.8	74.0	88.4	70.1
$\text{RM}_{\text{BT-BSR}} (\lambda = 10^{-4})$	70.0	61.4	52.9	95.1	47.9	74.3	87.4	69.9
$\text{RM}_{\text{BT-BSR}} (\lambda = 10^{-3})$	70.1	61.2	53.9	95.4	52.1	73.5	85.0	70.2
$\text{RM}_{\text{BT-BSR}} (\lambda = 10^{-2})$	70.0	60.1	53.1	96.0	55.7	72.5	81.9	70.0

set but in the general corpus: *i.e.*, overfitted to the train set.

Given a dataset \mathcal{D} with N rows, we collect a set of hidden states $H \in \mathbb{R}^{N \times H}$. To compute the effective rank of H , let $Q = \min(N, H)$, and denote the singular values of H by $\sigma_1, \sigma_2, \dots, \sigma_Q$, we have:

$$\text{erank}(H) = \exp \left(- \sum_{j=1}^Q \frac{\sigma_j}{\sum_{i=1}^Q \sigma_i} \log \frac{\sigma_j}{\sum_{i=1}^Q \sigma_i} \right). \quad (32)$$

In Table 1, we compute the erank of RM_{BT} and $\text{RM}_{\text{BT-BSR}}$ by collecting their hidden states for SKP ($\text{erank}_{\text{train}}$) and RM-Bench ($\text{erank}_{\text{eval}}$). While $\text{RM}_{\text{BT-BSR}}$ retains $\text{erank}_{\text{train}}$ in the evaluation set, $\text{RM}_{\text{BT-BSR}}$ experiences around 40% increase. Thus, RM_{BT} is prone to over-optimization, as previous analyses show. Furthermore, this makes RMs even more vulnerable to dataset biases, such as verbosity bias (Saito et al., 2023; Zhang et al., 2024b; Chen et al., 2024a).

BSR captures subtle preference factors When r^* is not available, we evaluate RMs through RM-Bench (Liu et al., 2025b), which provides preference pairs with subtle differences and achieves higher correlation against actual use cases than RewardBench (Lambert et al., 2025).

In Table 2, we observe RM’s accuracy consistently increasing in the response pairs with subtle differences (“Hard Acc”) as λ gets larger. Liu et al. (2025b) reports that *hard accuracy* of RM in RM-Bench has the highest correlation with the downstream policy’s actual performance, while the correlation was near zero for *easy accuracy*. Along with the fact that the over-confidence issue hinders accurate classification in hard tasks (Wei et al., 2022), we inspect that RM_{BT} fails to capture subtle differences in their representations, also supported by a significant increase in erank for unseen data in Table 1.

For final RLHF training, we select $\text{RM}_{\text{BT-BSR}}$ with $\lambda = 10^{-3}$ in Table 2, regarding a balance between the hard and easy tasks along with the overall scores. We report the reward logs during the training for both in Appendix D.

$\text{RM}_{\text{BT-BSR}}$ boosts AlpacaEval without verbosity We evaluate π_{BT} and $\pi_{\text{BT-BSR}}$ through length-controlled (LC) AlpacaEval 2.0 (Dubois et al., 2024), which GPT-4 (OpenAI

Table 3. AlpacaEval 2.0 average response length and length controlled win ate for the checkpoints throughout the RLHF process using Qwen2.5-1.5B with SFT on TULU3 mixture. Using $\text{RM}_{\text{BT+BSR}}$ significantly reduces verbosity while being best aligned.

Qwen2.5-1.5B	Length	LC AE2.0 (%)
SFT	2247	2.59
+ RLOO (RM_{BT})	2180	8.41
+ RLOO ($\text{RM}_{\text{BT+BSR}}$)	1337	9.02

et al., 2024) is used as LLM-as-a-judge. In Table 3, $\pi_{\text{BT-BSR}}$ demonstrates the highest LC win rate with a significantly shorter response than the SFT model and π_{BT} .

As preferring verbose response is a chronic problem in generative evaluators (Zheng et al., 2023; Dubois et al., 2024), reducing generation length by 40% compared to SFT model while achieving the best win rate in Table 3 highlights how $\text{RM}_{\text{BT-BSR}}$ could be advantageous for preventing policies falling into local minima while maximizing the reward: *i.e.*, reward gaming (Skalse et al., 2022; Pang et al., 2023; Chen et al., 2024b). Comprehensively, results underscore the effectiveness of $\mathcal{L}_{\text{BT-BSR}}$ for the overall RLHF pipeline.

6. Related Works

Over-optimization of reward models As formally outlined by Gao et al., 2023, *reward over-optimization* refers to the phenomenon where the reward model (RM) fails to generalize to the gold objective due to excessive training. To enhance out-of-distribution generalizability some works have proposed ensembling RMs (Gleave & Irving, 2022; Zhai et al., 2023; Coste et al., 2024), meta-learning RMs on the shifted target distribution (Wang et al., 2024a), augmenting preference data for RMs (Liu et al., 2025a), jointly learning the text-generation loss in reward modeling (Yang et al., 2024b), or constraining the preference optimization process itself (Moskovitz et al., 2024; Zhang et al., 2024a; Liu et al., 2024b; Gupta et al., 2025). Interestingly, Rafailov et al., 2024 outlined how a similar phenomenon can be seen in Direct Alignment Algorithms (DAAs), where the implicit rewards of the policy replace RM. On the other hand, some

applications on multilinguality (Wu et al., 2024; Hong et al., 2025) have suggested how RMs can possess cross-lingual transfer to languages unseen during reward modeling.

Evaluating reward models Parting from assessing the accuracy of the validation sets of popular preference datasets (Stiennon et al., 2020; Bai et al., 2022), Lambert et al., 2025 formalized a reproducible toolkit to evaluate reward models for enhanced explainability in diverse preference domains and tasks. Subsequent works have addressed improvements in cases of handling more sensitive, subtle cases (Liu et al., 2025b) and in wider coverage of real-world scenarios (Zhou et al., 2025) or specializations in tasks such as mathematical reasoning (Kim et al., 2024). Frick et al., 2024 introduced an evaluation pipeline directly targeted at assessing the RM in its role in the reinforcement learning with human feedback (RLHF) pipeline.

Conclusion

This study outlines why reward models (RMs) trained with the Bradley-Terry (BT) model loss can be vulnerable to over-optimization issues, losing generalizability to unseen tasks. We highlight the dispersion in the norm of hidden states in RMs as a primary source of such issue with theoretical analysis and empirical demonstrations across model families and sizes. Then, we propose batch-wise sum-to-zero regularization (**BSR**), an add-on to the BT model that penalizes the rewards for having an abnormally large magnitude. We present threefold experiments throughout the overall RLHF pipeline, starting by assessing the robustness of BSR and four baseline methods through the alignment against the synthetic gold preference model. Then, we RLHF training with RLOO using RMs trained with BT model and BSR, respectively. We observe a stronger alignment of the resulting policy after RLOO against the gold preference model. Eventually, we expand the experiments in 8B size model with high-quality preference data, where RM with BSR surpasses the state-of-the-art RM in 8B size with 7% improvements.

Impact Statement

This paper aims to identify the causes of reward model over-optimization and suggests a simple method to enhance robustness in reward models. As reward models are proxies for human preferences in language model alignment, the proposed method has the potential to impact our society, none of which we feel must be specifically highlighted here.

Acknowledgement

This work was supported by Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (RS-2024-

00398115, Technology research to ensure authenticity and consistency of results generated by AI) and (RS-2019-II190075, Artificial Intelligence Graduate School Program (KAIST)).

References

- Ahmadian, A., Cremer, C., Gallé, M., Fadaee, M., Kreutzer, J., Pietquin, O., Üstün, A., and Hooker, S. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL <https://aclanthology.org/2024.acl-long.662>.
- Allal, L. B., Lozhkov, A., Bakouch, E., Blázquez, G. M., Penedo, G., Tunstall, L., Marafioti, A., Kydlíček, H., Lajarín, A. P., Srivastav, V., Lochner, J., Fahlgrén, C., Nguyen, X.-S., Fourrier, C., Burtenshaw, B., Larcher, H., Zhao, H., Zakka, C., Morlon, M., Raffel, C., von Werra, L., and Wolf, T. Smollm2: When smol goes big – data-centric training of a small language model, 2025. URL <https://arxiv.org/abs/2502.02737>.
- Almazrouei, E., Alobeidli, H., Alshamsi, A., Cappelli, A., Cojocar, R., Debbah, M., Étienne Goffinet, Hesslow, D., Launay, J., Malartic, Q., Mazzotta, D., Noune, B., Pannier, B., and Penedo, G. The falcon series of open language models, 2023. URL <https://arxiv.org/abs/2311.16867>.
- Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., Das-Sarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.
- Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O’Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.
- Biš, D., Podkorytov, M., and Liu, X. Too much in common: Shifting of embeddings in transformer language models and its implications. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y. (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5117–5130, Online, June 2021. Association for Computa-

- tional Linguistics. doi: 10.18653/v1/2021.naacl-main.403. URL <https://aclanthology.org/2021.naacl-main.403/>.
- Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. ISSN 00063444. URL <http://www.jstor.org/stable/2334029>.
- Casper, S., Davies, X., Shi, C., Gilbert, T. K., Scheurer, J., Rando, J., Freedman, R., Korbak, T., Lindner, D., Freire, P., Wang, T. T., Marks, S., Segerie, C.-R., Carroll, M., Peng, A., Christoffersen, P. J., Damani, M., Slocum, S., Anwar, U., Siththaranjan, A., Nadeau, M., Michaud, E. J., Pfau, J., Krashennnikov, D., Chen, X., Langosco, L., Hase, P., Biyik, E., Dragan, A., Krueger, D., Sadigh, D., and Hadfield-Menell, D. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL <https://openreview.net/forum?id=bx24KpJ4Eb>. Survey Certification, Featured Certification.
- Chen, G. H., Chen, S., Liu, Z., Jiang, F., and Wang, B. Humans or LLMs as the judge? a study on judgement bias. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8301–8327, Miami, Florida, USA, November 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.474. URL <https://aclanthology.org/2024.emnlp-main.474/>.
- Chen, L., Zhu, C., Chen, J., Soselia, D., Zhou, T., Goldstein, T., Huang, H., Shoenybi, M., and Catanzaro, B. ODIN: Disentangled reward mitigates hacking in RLHF. In *Forty-first International Conference on Machine Learning*, 2024b. URL <https://openreview.net/forum?id=zcIV8OQFVF>.
- Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I., and Xing, E. P. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL <https://lmsys.org/blog/2023-03-30-vicuna/>.
- Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.
- Cortes, C. Support-vector networks. *Machine Learning*, 1995.
- Coste, T., Anwar, U., Kirk, R., and Krueger, D. Reward model ensembles help mitigate overoptimization. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=dcjtMYkpXx>.
- Cui, G., Yuan, L., Ding, N., Yao, G., He, B., Zhu, W., Ni, Y., Xie, G., Xie, R., Lin, Y., Liu, Z., and Sun, M. Ultrafeedback: boosting language models with scaled ai feedback. In *Proceedings of the 41st International Conference on Machine Learning, ICML’24*. JMLR.org, 2025.
- Davidson, R. R. On extending the bradley-terry model to accommodate ties in paired comparison experiments. *Journal of the American Statistical Association*, 65(329): 317–328, 1970. ISSN 01621459, 1537274X. URL <http://www.jstor.org/stable/2283595>.
- Ding, N., Chen, Y., Xu, B., Qin, Y., Hu, S., Liu, Z., Sun, M., and Zhou, B. Enhancing chat language models by scaling high-quality instructional conversations. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3029–3051, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.183. URL <https://aclanthology.org/2023.emnlp-main.183/>.
- Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Al-lonsius, D., Song, D., Pintz, D., Livshits, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Nail, G., Milalon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., and et al, I. K. The llama 3 herd of models, 2024. URL <https://arxiv.org/abs/2407.21783>.
- Dubois, Y., Liang, P., and Hashimoto, T. Length-controlled alpacaEval: A simple debiasing of automatic evaluators. In *First Conference on Language Modeling*, 2024. URL <https://openreview.net/forum?id=CybBmzWBX0>.
- Frick, E., Li, T., Chen, C., Chiang, W.-L., Angelopoulos, A. N., Jiao, J., Zhu, B., Gonzalez, J. E., and Stoica, I.

- How to evaluate reward models for rlhf. *arXiv preprint arXiv:2410.14872*, 2024.
- Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10835–10866. PMLR, 23–29 Jul 2023. URL <https://proceedings.mlr.press/v202/gao23h.html>.
- Gleave, A. and Irving, G. Uncertainty estimation for language reward models. *arXiv preprint arXiv:2203.07472*, 2022.
- Gupta, A., Tang, S., Song, Q., Zhu, S., Hong, J., Saha, A., Gupta, V., Lee, N., Kim, E., Zhu, J., Pillai, N., and Keerthi, S. S. AlphaPO - reward shape matters for LLM alignment. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://arxiv.org/abs/2501.03884>.
- Hong, J., Lee, N., and Thorne, J. ORPO: Monolithic preference optimization without reference model. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 11170–11189, Miami, Florida, USA, November 2024. Association for Computational Linguistics. URL <https://aclanthology.org/2024.emnlp-main.626>.
- Hong, J., Lee, N., Martínez-Castaño, R., Rodríguez, C., and Thorne, J. Cross-lingual transfer of reward models in multilingual alignment. In Chiruzzo, L., Ritter, A., and Wang, L. (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 82–94, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-190-2. URL <https://aclanthology.org/2025.naacl-short.8/>.
- Hsu, P.-L., Dai, Y., Kothapalli, V., Song, Q., Tang, S., Zhu, S., Shimizu, S., Sahni, S., Ning, H., and Chen, Y. Liger kernel: Efficient triton kernels for llm training, 2024. URL <https://arxiv.org/abs/2410.10989>.
- Hu, J., Wu, X., Zhu, Z., Xianyu, Wang, W., Zhang, D., and Cao, Y. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*, 2024.
- Huang, S., Noukhovitch, M., Hosseini, A., Rasul, K., Wang, W., and Tunstall, L. The n+ implementation details of RLHF with PPO: A case study on TL;DR summarization. In *First Conference on Language Modeling*, 2024. URL <https://openreview.net/forum?id=kH02ZTa8e3>.
- Huang, T.-k., Lin, C.-j., and Weng, R. A generalized bradley-terry model: From group competition to individual skill. In Saul, L., Weiss, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004. URL https://proceedings.neurips.cc/paper_files/paper/2004/file/825f9cd5f0390bc77c1fed3c94885c87-Paper.pdf.
- Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b, 2023a.
- Jiang, D., Ren, X., and Lin, B. Y. LLM-blender: Ensembling large language models with pairwise ranking and generative fusion. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14165–14178, Toronto, Canada, July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.792. URL <https://aclanthology.org/2023.acl-long.792/>.
- Kendall, M. *Rank Correlation Methods*. Griffin books on statistics. Hafner Publishing Company, 1962. URL <https://books.google.co.kr/books?id=lwhKAAAAMAAJ>.
- Kim, S., Kang, D., Kwon, T., Chae, H., Won, J., Lee, D., and Yeo, J. Evaluating robustness of reward models for mathematical reasoning, 2024. URL <https://arxiv.org/abs/2410.01729>.
- Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP ’23*, pp. 611–626, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702297. doi: 10.1145/3600006.3613165. URL <https://doi.org/10.1145/3600006.3613165>.
- Lambert, N., Morrison, J., Pyatkin, V., Huang, S., Ivison, H., Brahman, F., Miranda, L. J. V., Liu, A., Dziri, N., Lyu, S., Gu, Y., Malik, S., Graf, V., Hwang, J. D., Yang, J., Bras, R. L., Tafford, O., Wilhelm, C., Soldaini, L., Smith, N. A., Wang, Y., Dasigi, P., and Hajishirzi, H. Tulu 3: Pushing frontiers in open language model post-training, 2024. URL <https://arxiv.org/abs/2411.15124>.

- Lambert, N., Pyatkin, V., Morrison, J., Miranda, L. J. V., Lin, B. Y., Chandu, K., Dziri, N., Kumar, S., Zick, T., Choi, Y., Smith, N. A., and Hajishirzi, H. RewardBench: Evaluating reward models for language modeling. In Chiruzzo, L., Ritter, A., and Wang, L. (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1755–1797, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. URL <https://aclanthology.org/2025.findings-naacl.96/>.
- Liu, C. Y., Zeng, L., Liu, J., Yan, R., He, J., Wang, C., Yan, S., Liu, Y., and Zhou, Y. Skywork-Reward: Bag of Tricks for Reward Modeling in LLMs, October 2024a. URL <http://arxiv.org/abs/2410.18451>. arXiv:2410.18451 [cs].
- Liu, T., Xiong, W., Ren, J., Chen, L., Wu, J., Joshi, R., Gao, Y., Shen, J., Qin, Z., Yu, T., Sohn, D., Makarova, A., Liu, J. Z., Liu, Y., Piot, B., Ittycheriah, A., Kumar, A., and Saleh, M. RRM: Robust reward model training mitigates reward hacking. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL <https://openreview.net/forum?id=88AS5MQnmC>.
- Liu, Y., Yao, Z., Min, R., Cao, Y., Hou, L., and Li, J. RM-bench: Benchmarking reward models of language models with subtlety and style. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL <https://openreview.net/forum?id=QEHRmQPBdd>.
- Liu, Z., Lu, M., Zhang, S., Liu, B., Guo, H., Yang, Y., Blanchet, J., and Wang, Z. Provably mitigating overoptimization in RLHF: Your SFT loss is implicitly an adversarial regularizer. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL <https://openreview.net/forum?id=2cQ31PhkeO>.
- Meng, Y., Xia, M., and Chen, D. SimPO: Simple preference optimization with a reference-free reward. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=3Tzcot1LKb>.
- Moskovitz, T., Singh, A. K., Strouse, D., Sandholm, T., Salakhutdinov, R., Dragan, A., and McAleer, S. M. Confronting reward model overoptimization with constrained RLHF. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=gkfUvn0fLU>.
- Noukhovitch, M., Huang, S., Xhonneux, S., Hosseini, A., Agarwal, R., and Courville, A. Faster, more efficient RLHF through off-policy asynchronous learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=FhTAG591Ve>.
- OLMo, T., Walsh, P., Soldaini, L., Groeneveld, D., Lo, K., Arora, S., Bhagia, A., Gu, Y., Huang, S., Jordan, M., Lambert, N., Schwenk, D., Tafjord, O., Anderson, T., Atkinson, D., Brahman, F., Clark, C., Dasigi, P., Dziri, N., Guerquin, M., Ivison, H., Koh, P. W., Liu, J., Malik, S., Merrill, W., Miranda, L. J. V., Morrison, J., Murray, T., Nam, C., Pyatkin, V., Rangapur, A., Schmitz, M., Skjongsberg, S., Wadden, D., Wilhelm, C., Wilson, M., Zettlemoyer, L., Farhadi, A., Smith, N. A., and Hajishirzi, H. 2 olmo 2 furious, 2025. URL <https://arxiv.org/abs/2501.00656>.
- OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, J. H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk, Kondrich, A., Konstantinidis, A., Kosc, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O’Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M.,

- de Oliveira Pinto, H. P., Michael, Pokorný, Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M. B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. Gpt-4 technical report, 2024. URL <https://arxiv.org/abs/2303.08774>.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- Pang, R. Y., Padmakumar, V., Sellam, T., Parikh, A., and He, H. Reward gaming in conditional text generation. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4746–4763, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.262. URL <https://aclanthology.org/2023.acl-long.262/>.
- Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=HPuSIXJaa9>.
- Rafailov, R., Chittepudi, Y., Park, R., Sikchi, H., Hejna, J., Knox, W. B., Finn, C., and Niekum, S. Scaling laws for reward model overoptimization in direct alignment algorithms. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=pf4OuJyn4Q>.
- Rajbhandari, S., Rasley, J., Ruwase, O., and He, Y. Zero: memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC ’20. IEEE Press, 2020. ISBN 9781728199986.
- Roy, O. and Vetterli, M. The effective rank: A measure of effective dimensionality. In *2007 15th European Signal Processing Conference*, pp. 606–610, 2007.
- Saito, K., Wachi, A., Wataoka, K., and Akimoto, Y. Verbosity bias in preference labeling by large language models. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023. URL <https://openreview.net/forum?id=magEgFpKly>.
- Skalse, J. M. V., Howe, N. H. R., Krashenninnikov, D., and Krueger, D. Defining and characterizing reward gaming. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL <https://openreview.net/forum?id=yb3HOXO3lX2>.
- Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.
- Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.
- Team, G., Riviere, M., Pathak, S., Sessa, P. G., Hardin, C., Bhupatiraju, S., Hussenot, L., Mesnard, T., Shahriari, B., Ramé, A., Ferret, J., Liu, P., Tafti, P., Friesen, A., Casbon, M., Ramos, S., Kumar, R., Lan, C. L., Jerome, S., Tsitsulin, A., Vieillard, N., Stanczyk, P., Girgin, S., Momchev, N., Hoffman, M., Thakoor, S., Grill, J.-B., Neyshabur, B., Bachem, O., Walton, A., Severyn, A., Parrish, A., Ahmad, A., Hutchison, A., Abdagic, A., Carl, A., Shen, A., Brock, A., Coenen, A., Laforge, A., Paterson, A., Bastian, B., Piot, B., Wu, B., Royal, B., Chen, C., Kumar, C., Perry, C., Welty, C., Choquette-Choo, C. A., Sinopalnikov, D., Weinberger, D., Vijaykumar, D., Rogozińska, D., Heribson, D., Bandy, E., Wang, E., Noland, E., Moreira, E., Senter, E., Eltyshev, E., Visin, F., Rasskin, G., Wei, G., Cameron, G., Martins, G., Hashemi, H., Klimczak-Plucińska, H., Batra, H., Dhand, H., Nardini, I., Mein, J., Zhou, J., Svensson, J., Stanway, J., Chan, J., Zhou, J. P., Carrasqueira, J., Iljazi, J., Becker, J., Fernandez, J., van Amersfoort, J., Gordon, J., Lipschultz, J., Newlan, J., yeong Ji, J., Mohamed, K., Badola, K., Black, K., Millican, K., McDonnell, K., Nguyen, K., Sodhia, K., Greene, K., Sjoesund, L. L., Usui, L., Sifre, L., Heuermann, L., Lago, L., McNealus, L., Soares, L. B., Kilpatrick, L., Dixon, L., Martins, L., Reid, M., Singh, M., Iverson,

- M., Görner, M., Velloso, M., Wirth, M., Davidow, M., Miller, M., Rahtz, M., Watson, M., Risdal, M., Kazemi, M., Moynihan, M., Zhang, M., Kahng, M., Park, M., Rahman, M., Khatwani, M., Dao, N., Bardoliwalla, N., Devanathan, N., Dumai, N., Chauhan, N., Wahltinez, O., Botarda, P., Barnes, P., Barham, P., Michel, P., Jin, P., Georgiev, P., Culliton, P., Kuppala, P., Comanescu, R., Merhej, R., Jana, R., Rokni, R. A., Agarwal, R., Mullins, R., Saadat, S., Carthy, S. M., Perrin, S., Arnold, S. M. R., Krause, S., Dai, S., Garg, S., Sheth, S., Ronstrom, S., Chan, S., Jordan, T., Yu, T., Eccles, T., Hennigan, T., Kocisky, T., Doshi, T., Jain, V., Yadav, V., Meshram, V., Dharmadhikari, V., Barkley, W., Wei, W., Ye, W., Han, W., Kwon, W., Xu, X., Shen, Z., Gong, Z., Wei, Z., Cotruta, V., Kirk, P., Rao, A., Giang, M., Peran, L., Warkentin, T., Collins, E., Barral, J., Ghahramani, Z., Hadsell, R., Sculley, D., Banks, J., Dragan, A., Petrov, S., Vinyals, O., Dean, J., Hassabis, D., Kavukcuoglu, K., Farabet, C., Buchatskaya, E., Borgeaud, S., Fiedel, N., Joulin, A., Kenealy, K., Dadashi, R., and Andreev, A. Gemma 2: Improving open language models at a practical size, 2024. URL <https://arxiv.org/abs/2408.00118>.
- Team, M. N. Introducing mpt-30b: Raising the bar for open-source foundation models. *Blog*, 2023. www.mosaicml.com/blog/mpt-30b.
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models, 2023.
- Tunstall, L., Lambert, N., Rajani, N., Beeching, E., Le Scao, T., von Werra, L., Han, S., Schmid, P., and Rush, A. Creating a coding assistant with starcoder. *Hugging Face Blog*, 2023. <https://huggingface.co/blog/starchat>.
- Tunstall, L., Beeching, E. E., Lambert, N., Rajani, N., Rasul, K., Belkada, Y., Huang, S., Werra, L. V., Fourrier, C., Habib, N., Sarrazin, N., Sanseviero, O., Rush, A. M., and Wolf, T. Zephyr: Direct distillation of LM alignment. In *First Conference on Language Modeling*, 2024. URL <https://openreview.net/forum?id=aKkAwZB6JV>.
- von Werra, L., Belkada, Y., Tunstall, L., Beeching, E., Thrush, T., Lambert, N., Huang, S., Rasul, K., and Gallouédec, Q. Trl: Transformer reinforcement learning. <https://github.com/huggingface/trl>, 2020.
- Wang, B., Zheng, R., Chen, L., Liu, Y., Dou, S., Huang, C., Shen, W., Jin, S., Zhou, E., Shi, C., Gao, S., Xu, N., Zhou, Y., Fan, X., Xi, Z., Zhao, J., Wang, X., Ji, T., Yan, H., Shen, L., Chen, Z., Gui, T., Zhang, Q., Qiu, X., Huang, X., Wu, Z., and Jiang, Y.-G. Secrets of rlhf in large language models part ii: Reward modeling, 2024a. URL <https://arxiv.org/abs/2401.06080>.
- Wang, H., Xiong, W., Xie, T., Zhao, H., and Zhang, T. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10582–10592, Miami, Florida, USA, November 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.620. URL <https://aclanthology.org/2024.findings-emnlp.620/>.
- Wei, H., Xie, R., Cheng, H., Feng, L., An, B., and Li, Y. Mitigating neural network overconfidence with logit normalization. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23631–23644. PMLR, 17–23 Jul 2022. URL <https://proceedings.mlr.press/v162/wei22d.html>.
- Wei, L., Tan, Z., Li, C., Wang, J., and Huang, W. Diff-erank: A novel rank-based metric for evaluating large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=nvn80cscVm>.
- Wu, Z., Balashankar, A., Kim, Y., Eisenstein, J., and Beirami, A. Reuse your rewards: Reward model transfer for zero-shot cross-lingual alignment. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1332–1353, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.79. URL <https://aclanthology.org/2024.emnlp-main.79/>.
- Xu, C., Sun, Q., Zheng, K., Geng, X., Zhao, P., Feng, J., Tao, C., Lin, Q., and Jiang, D. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Confer-*

- ence on Learning Representations, 2024. URL <https://openreview.net/forum?id=CfXh93NDgH>.
- Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., Dong, G., Wei, H., Lin, H., Tang, J., Wang, J., Yang, J., Tu, J., Zhang, J., Ma, J., Yang, J., Xu, J., Zhou, J., Bai, J., He, J., Lin, J., Dang, K., Lu, K., Chen, K., Yang, K., Li, M., Xue, M., Ni, N., Zhang, P., Wang, P., Peng, R., Men, R., Gao, R., Lin, R., Wang, S., Bai, S., Tan, S., Zhu, T., Li, T., Liu, T., Ge, W., Deng, X., Zhou, X., Ren, X., Zhang, X., Wei, X., Ren, X., Liu, X., Fan, Y., Yao, Y., Zhang, Y., Wan, Y., Chu, Y., Liu, Y., Cui, Z., Zhang, Z., Guo, Z., and Fan, Z. Qwen2 technical report, 2024a. URL <https://arxiv.org/abs/2407.10671>.
- Yang, R., Ding, R., Lin, Y., Zhang, H., and Zhang, T. Regularizing hidden states enables learning generalizable reward model for LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL <https://openreview.net/forum?id=jwh9MHEfmY>.
- Yuan, L., Cui, G., Wang, H., Ding, N., Wang, X., Deng, J., Shan, B., Chen, H., Xie, R., Lin, Y., Liu, Z., Zhou, B., Peng, H., Liu, Z., and Sun, M. Advancing LLM reasoning generalists with preference trees. In *AI for Math Workshop @ ICML 2024*, 2024. URL <https://openreview.net/forum?id=2Y1iiCqM5y>.
- Zhai, Y., Zhang, H., Lei, Y., Yu, Y., Xu, K., Feng, D., Ding, B., and Wang, H. Uncertainty-penalized reinforcement learning from human feedback with diverse reward lora ensembles. *arXiv preprint arXiv:2401.00243*, 2023.
- Zhang, X., Ton, J.-F., Shen, W., Wang, H., and Liu, Y. Mitigating reward overoptimization via lightweight uncertainty estimation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a. URL <https://openreview.net/forum?id=kYio3xH6eb>.
- Zhang, X., Xiong, W., Chen, L., Zhou, T., Huang, H., and Zhang, T. From lists to emojis: How format bias affects model alignment, 2024b. URL <https://arxiv.org/abs/2409.11704>.
- Zhao, Y., Gu, A., Varma, R., Luo, L., Huang, C.-C., Xu, M., Wright, L., Shojanazeri, H., Ott, M., Shleifer, S., Desmaison, A., Balioglu, C., Damania, P., Nguyen, B., Chauhan, G., Hao, Y., Mathews, A., and Li, S. Pytorch fsdp: Experiences on scaling fully sharded data parallel. *Proc. VLDB Endow.*, 16(12):3848–3860, 2023. URL <https://www.vldb.org/pvldb/vol16/p3848-huang.pdf>.
- Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL <https://openreview.net/forum?id=uccHPGDlao>.
- Zhou, E., Zheng, G., Wang, B., Xi, Z., Dou, S., Bao, R., Shen, W., Xiong, L., Fan, J., Mou, Y., Zheng, R., Gui, T., Zhang, Q., and Huang, X. RMB: Comprehensively benchmarking reward models in LLM alignment. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=kmgrlG9TR0>.
- Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences, 2020.

A. Training Configurations

For supervised fine-tuning (SFT) and reward modeling, we use Liger-Kernel (Hsu et al., 2024) with DeepSpeed ZeRO-3 (Rajbhandari et al., 2020) and FSDP (Zhao et al., 2023) for efficient training. Including reinforcement learning with human feedback (RLHF) phase, we utilize the TRL library (von Werra et al., 2020) to adjust to our usage. We used NVIDIA A100 and A6000 GPUs throughout the experiments.

A.1. Supervised fine-tuning

We train every model on UltraChat for a single epoch with a global batch size of 512 following Tunstall et al. (2024). We set a learning rate of 10^{-5} with 10% warmup and cosine decay.

A.2. Reward modeling

We train reward models on top of the SFT models above with four different seeds. We fix the global batch size of 128 across the models and methods. We set a learning rate of 5×10^{-6} for Llama-3.2-1B and Qwen2.5-1.5B models, 3×10^{-6} for 3B models, and 2×10^{-6} for Llama-3.1-8B and Qwen2.5-7B models. 5% warmup and linear decay were applied following Lambert et al. (2024). We use FSDP for distributed training in reward modeling. We set $\lambda = 10^{-3}$ in Section 5.1 and ablate different λ in Section 5.3.

A.3. Reinforcement learning with human feedback

We leverage Async RLHF (Noukhovitch et al., 2025) with vLLM (Kwon et al., 2023) as a backend to reduce the bottleneck in on-policy generations. To test reward models with varying sizes under controlled training configurations (*e.g.*, batch size), we separately deploy the reward models following OpenRLHF (Hu et al., 2024). We use a batch size of 1 for every reward model to prevent potential distortions in reward scores with padding applied.

Table 4. RLOO training configuration details for each section. We train Qwen2.5-1.5B with SFT using corresponding reward models for each section using 4 A6000 GPUs, excluding the GPUs assigned for the reward model and vLLM engine for on-policy generation.

Category	Section 3.2	Section 3.3
Learning Rate	2×10^{-6}	1×10^{-6}
β	0.05	0.05
Number of responses (k)	2	2
Global Batch (Effective)	128	128
Learning Rate Scheduler	Linear Decay	Linear Decay
Warmup Ratio	0.03	0.03
Training Epochs	5	5

B. Data Preprocessing

We adopt UltraFeedback³ (Cui et al., 2025) for our experiments⁴. Before splitting the train and validation sets, we filtered items with duplicated prompts and responses. We first removed the items with identical prompts and removed the rows when at least two responses out of four were identical. As a result, we used 62,282 rows after filtration from 63,966.

In detail, 17 models comprising $\mathcal{M}_{\text{train}}$ for UF are: GPT-3.5-Turbo, GPT-4 (OpenAI et al., 2024), Bard⁵, Llama-2-7B-Chat, Llama-2-13B-Chat, Llama-2-70B-Chat (Touvron et al., 2023), WizardLM-7B, WizardLM-13B, WizardLM-70B (Xu et al., 2024), Vicuna-33B (Chiang et al., 2023), Alpaca-7B (Taori et al., 2023), Falcon-40B-Instruct (Almazrouei et al., 2023), MPT-30B-Chat (Team, 2023), StarChat-beta (Tunstall et al., 2023), and Pythia-12B (Biderman et al., 2023).

Since large proportion of the models in $\mathcal{M}_{\text{train}}$ are Llama-2 based models, we selected distinct model families for $\mathcal{M}_{\text{valid}}$, including OLMO2-7B-Instruct (OLMo et al., 2025), SmolLM2-1.7B-Instruct (Allal et al., 2025), Mistral-Instruct-v0.2 (Jiang et al., 2023a), and Gemma-2-2B-It (Team et al., 2024).

Train set ($\mathcal{D}_{\text{train}}$) First, we select a random set of 51,200 samples from UF as the *train set*. Then, we choose two random responses out of four for each prompt in the train set. Thereby, we have 51,200 triplets comprising prompt and corresponding chosen and rejected responses, according to ArmoRM.

Validation 1 - in-domain (\mathcal{D}_{ID}) Then, we use the remaining two responses in 51,200 prompts in the train set as the *in-domain validation set* to evaluate if the trained reward models can generalize in **same prompt and response spaces**. Since we have two responses per prompt, we use binary accuracy as an evaluation metric.

Validation 2 - prompt out-of-domain ($\mathcal{D}_{\text{Prompt-ODD}}$) We set the remaining 12,800 instances as a *prompt out-of-domain (Prompt OOD)* validation set, having **different prompt space but same response space**. Since we have four responses per prompt, we use Kendall’s τ ranking correlation (Kendall, 1962) as an evaluation metric.

Validation 3 - response out-of-domain ($\mathcal{D}_{\text{Response-ODD}}$) For the prompts in the train set, we additionally generate four different responses from the new models: Gemma-2-2B-It (Team et al., 2024), Olmo2-7B-Instruct⁶ (OLMo et al., 2025), SmolLM2-1.7B-Instruct (Allal et al., 2025), and Mistral-Instruct-v0.2 (Jiang et al., 2023a). By having *response out-of-domain (Response OOD)* validation set, we assess reward models when **prompt space stays the same, but the response space differs**. Since we have four responses per prompt, we use Kendall’s τ as an evaluation metric.

Validation 4 - mutual out-of-domain ($\mathcal{D}_{\text{Mutual-ODD}}$) Using the same models in the Response OOD set, we generate the responses for the prompts from the Prompt OOD set, having *mutual out-of-domain (Mutual OOD)* validation set. Here, we test the model’s robustness when **both the prompt and response spaces are distinct**. Since we have four responses per prompt, we use Kendall’s τ as an evaluation metric.

³<https://huggingface.co/datasets/openbmb/UltraFeedback>

⁴The final dataset can be found in: <https://huggingface.co/rm-robustness>

⁵<https://blog.google/technology/ai/bard-google-ai-search-updates/>

⁶<https://huggingface.co/allenai/OLMo-2-1124-7B-Instruct>

C. Robustness Assessment with Gold Preference Model

We report the full results of visualizing Figure 5 in Table 5.

Table 5. Preference prediction accuracy over different types of validation sets.

Model	Method	In-Domain (Accuracy \uparrow)	Prompt OOD (Kendall's $\tau \uparrow$)	Response OOD (Kendall's $\tau \uparrow$)	Mutual OOD (Kendall's $\tau \uparrow$)
Llama-3.2 (1B)	\mathcal{L}_{BT} (Bradley & Terry, 1952)	0.8132	0.6157	0.5115	0.5065
	$\mathcal{L}_{BT-Hinge}$ (Cortes, 1995)	0.8111	0.6102	0.5168	0.5131
	$\mathcal{L}_{BT-Norm}$ (Wei et al., 2022)	0.7792	0.5580	0.4503	0.4488
	\mathcal{L}_{BT-DR} (Yuan et al., 2024)	0.8029	0.6015	0.493	0.4937
	\mathcal{L}_{BT-BSR} (Ours)	0.813	0.6198	0.5246	0.5206
Qwen2.5 (1.5B)	\mathcal{L}_{BT} (Bradley & Terry, 1952)	0.8415	0.6835	0.5601	0.5566
	$\mathcal{L}_{BT-Hinge}$ (Cortes, 1995)	0.8388	0.6756	0.5592	0.5561
	$\mathcal{L}_{BT-Norm}$ (Wei et al., 2022)	0.8177	0.6403	0.4502	0.4507
	\mathcal{L}_{BT-DR} (Yuan et al., 2024)	0.8326	0.6625	0.5187	0.5201
	\mathcal{L}_{BT-BSR} (Ours)	0.8423	0.6849	0.5618	0.5573
Llama-3.2 (3B)	\mathcal{L}_{BT} (Bradley & Terry, 1952)	0.8478	0.679	0.6195	0.609
	$\mathcal{L}_{BT-Hinge}$ (Cortes, 1995)	0.8458	0.6829	0.6210	0.61
	$\mathcal{L}_{BT-Norm}$ (Wei et al., 2022)	0.8161	0.6272	0.5542	0.5509
	\mathcal{L}_{BT-DR} (Yuan et al., 2024)	0.8375	0.6638	0.5916	0.5849
	\mathcal{L}_{BT-BSR} (Ours)	0.8491	0.702	0.6402	0.6306
Qwen2.5 (3B)	\mathcal{L}_{BT} (Bradley & Terry, 1952)	0.8496	0.705	0.5917	0.587
	$\mathcal{L}_{BT-Hinge}$ (Cortes, 1995)	0.8488	0.6984	0.5784	0.5685
	$\mathcal{L}_{BT-Norm}$ (Wei et al., 2022)	0.8272	0.6605	0.5295	0.5282
	\mathcal{L}_{BT-DR} (Yuan et al., 2024)	0.8446	0.6855	0.5602	0.5564
	\mathcal{L}_{BT-BSR} (Ours)	0.8541	0.7202	0.5991	0.6106

D. Training logs for Section 5.3

We report additional training logs for RLOO training with Skywork-Reward-Llama-3.1-8B-v0.2 (RM_{BT}) and $RM_{BT-BSR}(\lambda = 10^{-3})$ based on Llama-3.1-8B in Table 2.

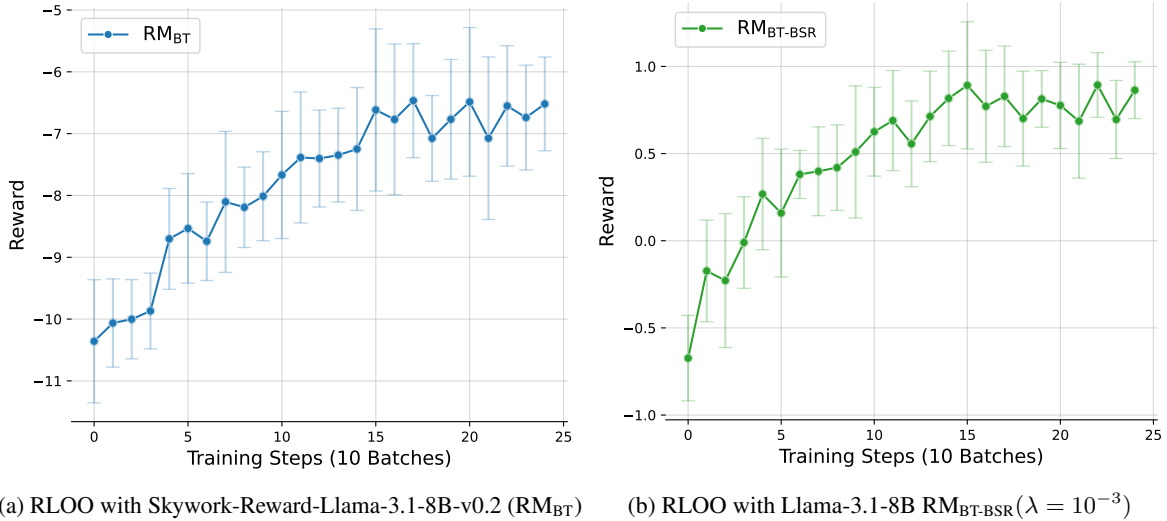


Figure 6. Reward maximization trajectories with Skywork-Reward-Llama-3.1-8B-v0.2 (RM_{BT}) and Llama-3.1-8B RM_{BT-BSR} .