Bias Fitting to Mitigate Length Bias of Reward Model in RLHF

Kangwen Zhao[1], Jianfeng Cai[1], Jinhua Zhu[1]
Ruopei Sun[1], Dongyun Xue[1], Wengang Zhou[1*], Li Li[1], Houqiang Li[1*]
[1]University of Science and Technology of China
{zkwzkw,xiaobaicai,teslazhu}@mail.ustc.edu.cn
{ruopeisun,andyxue}@mail.ustc.edu.cn,{zhwg,lil1,lihq}@ustc.edu.cn

Abstract

Reinforcement Learning from Human Feedback (RLHF) relies on reward models to align large language models with human preferences. However, RLHF often suffers from reward hacking, wherein policy learning exploits flaws in the trained reward model to maximize reward scores without genuinely aligning with human preferences. A significant example of such reward hacking is length bias, where reward models usually favor longer responses irrespective of actual response quality. Previous works on tackling length bias have notable limitations, these approaches either mitigate bias without characterizing the bias form, or simply assume a linear length-reward relation. To accurately model the intricate nature of length bias and facilitate more effective bias mitigation, we propose FiMi-RM (Bias **Fi**tting to **Mi**tigate Length Bias of **R**eward **M**odel), a framework that autonomously learns and corrects underlying bias patterns. Our approach consists of three stages: First, we warm up by training a standard reward model which inherently contains length bias. Next, we deploy a lightweight fitting model to capture the non-linear relation between length and reward. Finally, we incorporate this learned relation into the reward model, effectively decoupling length from reward while preserving preference modeling capabilities. Experimental results demonstrate that FiMi-RM achieves a more balanced length-reward distribution. Furthermore, when applied to alignment algorithms such as Direct Preference Optimization (DPO) and Best-of-N (BoN), our debiased reward model improves length-controlled win rate and reduces verbosity without compromising its performance.

1 Introduction

Reinforcement Learning from Human Feedback (RLHF) (Askell et al., 2021; Ouyang et al., 2022; Ziegler et al., 2020; Dong et al., 2024) is a popular method for aligning large language models (LLMs) with human preferences, used in models like GPT (OpenAI et al., 2024), Qwen (Qwen et al., 2025; Yang et al., 2024), DeepSeek (DeepSeek-AI et al., 2025b,a), Gemini (Team et al., 2024a) and Llama (Grattafiori et al., 2024; Touvron et al., 2023). The framework involves three stages: supervised finetuning, reward model training via comparisons between preferred and dispreferred outputs (using methods like Bradley-Terry model (Bradley and Terry, 1952)), and reinforcement learning optimization (Schulman et al., 2017).

However, RLHF generally suffers from reward hacking (Gao et al., 2023; Weng, 2024), where policy learning leverages flaws in the trained reward model to maximize reward scores but does not learn the true human preferences. Empirical analysis reveals that reward hacking manifests through multiple mechanisms: (1) Explicit surface-level bias, such as reward model usually favoring longer responses (Singhal et al., 2023) or preferring particular response formats (*e.g.*, numbered lists or markdown tables) (Zhang et al., 2024); (2) Implicit semantic bias, which arises from latent correlations in the training data distribution, where the reward model learns to associate higher reward with specific syntactic structures or topic distributions that match frequent patterns in the preference dataset (Pang et al., 2023; OpenAI, 2025).

A particularly prevalent form of reward hacking is length bias (Singhal et al., 2023; Köpf et al., 2023), where reward models favor longer outputs over shorter ones. This bias not only distorts the reward model's preference modeling but also leads to excessively verbose generations in reinforcement learning finetuned models. A key factor of this problem lies in human preference data, which often exhibits bias and inconsistencies due to challenges such as imperfect rating criteria and variability in annotator quality (Chen et al., 2024b; Stiennon et al., 2020; Pang et al., 2023; Lambert and Calan-

*Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2912–2927

dra, 2023). Specifically, with respect to length, human raters disproportionately favor longer outputs, a tendency that reward models can exploit, thereby causing length bias. Given the inherent difficulties in obtaining perfectly reliable human annotations, developing algorithmic approaches to mitigate such spurious correlations becomes increasingly crucial.

Existing approaches sometimes do not characterize the bias form. For instance, RRM (Liu et al., 2025) adopts a causal framework to achieve a more balanced data distribution, while other methods incorporate KL regularization terms during policy training (Stiennon et al., 2020; Ouyang et al., 2022). Alternatively, other studies assume a linear length-reward relation for tractability. ODIN (Chen et al., 2024b), for example, introduces a dual-headed architecture designed to decouple length-dependent scoring from quality-based assessment, and uses the Pearson correlation coefficient (Pearson, 1895) to quantify the length-reward relation. Similarly, length penalty (Singhal et al., 2023) directly subtracts the product of length and a constant from the reward to mitigate bias. Additionally, Huang *et.al.* (Huang et al., 2025) calculate the hacked reward by performing linear regression on points within a certain neighborhood during the reward model inference phase. Although the linear assumption offers mathematical simplicity and intuitive feasibility, it fails to capture some details, like non-linear features where length interacts with reward in complex ways.

To overcome this problem, we introduce FiMi-RM, an automated framework designed to model the complex non-linear relation between output length and reward scores, enabling more precise debiasing. The method begins by training a conventional reward model which inherently has length-related bias to warm up. Building upon this, a lightweight fitting model is trained to explicitly characterize how reward scores correlate with response length. By integrating these learned patterns into the reward model, the system effectively mitigates length bias without compromising its core preference modeling functionality. Empirical results confirm that FiMi-RM achieves a more balanced length-reward distribution. When deployed in downstream algorithms such as Direct Preference Optimization (DPO) (Rafailov et al., 2023) and Best-of-N (BoN) (Gui et al., 2024; Sessa et al., 2025; Dong et al., 2023), the debiased model demonstrates improved performance on length-controlled win rates, reducing excessive verbosity while maintaining competitive task accuracy. Further analysis of the fitting process reveals a multi-stage bias pattern: exhibiting strong linear correlation for short responses, flattening of the relation for longer responses and in some cases a slight downward tendency for extended outputs. Our contributions can be summarized as:

- We propose a multi-stage framework that autonomously learns non-linear relation between response length and hacked reward and uses this relation to better mitigate the length bias.
- We demonstrate the effectiveness of our length debiasing approach through comprehensive validation, including length-reward distribution on preference dataset, length-controlled win rate and length distribution of responses selected by reward models.
- We show the fitting result of the relation between length and hacked reward and identify that the length bias in the reward model is non-linear, which further illustrates the importance of debiasing with non-linear relations.

2 Related Work

Reinforcement Learning From Human Feedback RLHF (Askell et al., 2021; Ouyang et al., 2022; Ziegler et al., 2020; Dong et al., 2024) is an optimization algorithm proposed to align with human preferences. This algorithm is often diverse, with the most basic one being PPO (Schulman et al., 2017). Building upon PPO, several improved methods have been derived: GRPO (Shao et al., 2024) optimizes strategies through relative reward comparisons among multiple candidate outputs within a group, eliminating the need for a separate value model; DAPO (Yu et al., 2025) addresses issues such as entropy collapse, reward noise, and training instability in GRPO; and BoN (Gui et al., 2024; Sessa et al., 2025; Dong et al., 2023) directly utilizes the reward model to select the one with the highest reward score as the final output. Additionally, DPO (Rafailov et al., 2023) unifies reward modeling and reinforcement learning into a single stage by directly optimizing the policy on pairwise preference data, eliminating the need for an explicit reward model and online sampling. Simpo (Meng et al., 2024) further enhances model performance by removing the reference model and incorporating target reward boundaries and length normalization on the basis of DPO. Apart from them, there are many excellent studies that have contributed to the RLHF (Chen et al., 2024a; Hong et al., 2024; Gheshlaghi Azar et al., 2024; Ethayarajh et al., 2024; Richemond et al., 2024).

Length Bias in Reward Hacking A typical example of reward hacking is length bias, where the reward model prefers longer responses irrespective of actual response quality, leading the trained policy to generate unnecessarily verbose outputs. A part of existing approaches alleviate length bias through comprehensive reward hacking mitigation strategies, like some incorporate KL regularization terms during policy training (Stiennon et al., 2020; Ouyang et al., 2022). Additionally, Eisenstein *et.al.* (Eisenstein et al., 2024) point out that reward model ensembles can alleviate reward hacking and WARP (Ramé et al., 2024a) as well as WARM (Ramé et al., 2024b) utilize model merging techniques to reduce reward hacking, whereas RRM (Liu et al., 2025) introduces a data augmentation approach by incorporating a causal framework to alleviate the hacking. Apart from these, others specifically target length debiasing, such as length penalty (Singhal et al., 2023) directly subtracts the product of length and a certain coefficient from the reward to debias in a simple and intuitive way. Shen *et.al.* (Shen et al., 2023) applying Product-of-Experts to decouple the length and reward. Huang *et.al.* (Huang et al., 2025) derive the hacked reward by applying linear regression to nearby points in the reward model's inference stage. Park *et al.* (Park et al., 2024) introduce a length regularization term into the DPO objective, and SamPO (Lu et al., 2024) address the cumulative length-dependent structure of KL divergence through per-token normalization in DPO. Moreover, ODIN (Chen et al., 2024b) decoupling the reward model's scoring for length and quality with two-head structure to mitigate length bias. ALBM (Bu et al., 2025) further introduces a prompt analyzer that adaptively reweights the reward based on the prompt. These works either do not explicitly model the form of length bias or directly assume a linear relation between them. Therefore in this paper we continue to focus on length debiasing but moves beyond the simplistic assumption of a linear relation between length and reward bias. Instead, we employ a dedicated lightweight model to directly fit this relation, enabling more precise length debiasing based on an accurate understanding of the bias.

3 Method

This section elaborates on our whole framework as shown in Figure 1(a), which consists of three key stages: (1) a warm-up stage, where a standard reward model is trained and retains its inherent length bias, (2) a bias-fitting stage, where a lightweight model learns the relation between response length and biased reward, thereby capturing non-linear bias patterns, and (3) a debiasing stage, where the fitted relation is incorporated to debias.

Additionally, our framework employs two distinct models: (1) Reward model ($model_r(x, y)$ or $r(x, y)$ for simplicity), which serves as a reward function for response quality. This model is initialized from an existing large language models. (2) Fitting model ($model_f(y)$), a lightweight model designed to fit the length bias inherent in the reward model $r(x, y)$.

3.1 Warm-Up

The primary objective of the warm-up stage is to obtain a reward model with inherent length bias, a systematic tendency to assign a higher score to longer responses before implementing corrective measures. We initialize training using the standard reward modeling paradigm based on the Bradley-Terry model (Bradley and Terry, 1952), where given an input prompt x, the human-preferred response is denoted as y_w and the dispreferred response as y_l. The training loss function is:

$$\mathcal{L}_{\text{BT}} = -\mathbb{E}_{(x,y_w,y_l)}\left[\log\sigma\left(r(x, y_w) - r(x, y_l)\right)\right]. \tag{1}$$

Previous approaches operating under the assumption of an approximate linear relation between response length and reward, so they could directly apply length debiasing during initial training. In contrast, our method requires a precise characterization of this relation. The warm-up stage deliberately preserves length bias to enable subsequent learning and systematic removal of this bias.

3.2 Length Bias Fitting

After training a reward model with inherent length bias, we proceed to formally characterize and mitigate this relation using the fitting model. The proposed approach operates as follows: given the input response length $\text{len}(y)$, we first project this scalar value into a d-dimensional features (in our training d equals to 32). Inspired by positional encoding (PE) (Vaswani et al., 2017), we treat the response length as the position in PE, thereby deriving a

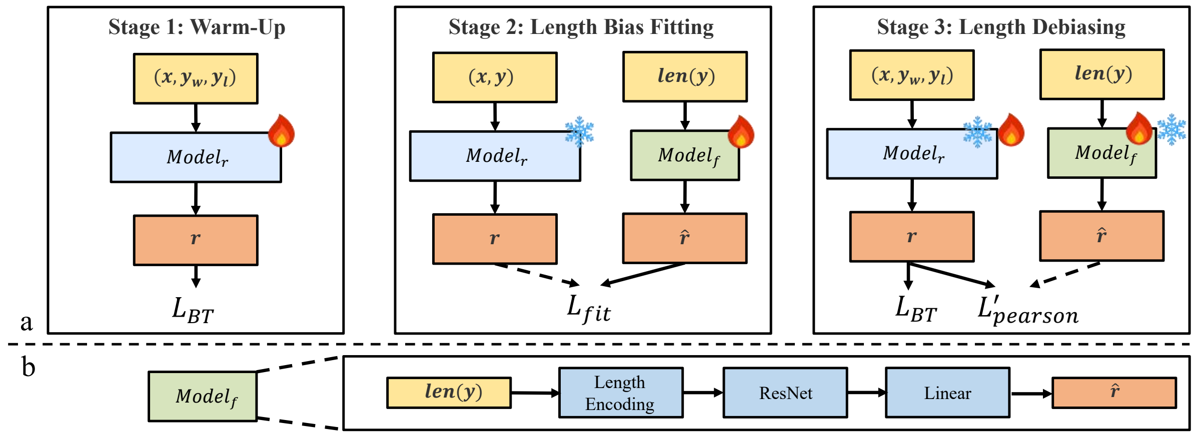

Figure 1: **a:** An overview of our method. We first use traditional reward model training to initially establish the model's length bias. Then we employ a lightweight fitting model to fit the reward hacking: given the length of a response, we minimize loss to make the output of the fitting model as close as possible to that of the reward model. The final step involves debiasing the length bias in the reward model based on the relation fitted by the fitting model, these two models take turns to train. **b:** The detailed architecture of the $model_f$.

length encoding (LE) that embeds length information into features. Subsequently, the features after encoding are then processed through ResNet (He et al., 2016) architecture, the output is then fed into a final linear projection layer which serves as the regression head to produce the predicted reward $\hat{r}$ (Figure 1(b)). The optimization objective of the $model_f$ is to minimize the discrepancy between the predicted reward $\hat{r}$ and the actual reward output r. We formulate the loss function as:

$$\mathcal{L}_{fit} = -\mathcal{L}_{pearson}, \mathcal{L}_{pearson} = \left|\rho\left(r_{sg}, \hat{r}\right)\right|, \quad (2)$$

$$\hat{r} = model_f(\text{len}(y)), \; r_{sg} = model_r(x, y). \quad (3)$$

The $\rho\left(r_{sg}, \hat{r}\right)$ in Equation 2 means the Pearson correlation coefficient (Pearson, 1895), which could be calculated by:

$$\rho(r, \hat{r}) = \frac{\sum_{i=1}^{n}(r_i - \bar{r})(\hat{r}_i - \bar{\hat{r}})}{\sqrt{\sum_{i=1}^{n}(r_i - \bar{r})^2}\sqrt{\sum_{i=1}^{n}(\hat{r}_i - \bar{\hat{r}})^2}}. \quad (4)$$

Here r_i denotes the reward assigned by the reward model to the i-th sample within the current batch, and $\bar{r}$ is the corresponding batch mean; analogously, $\hat{r}_i$ and $\bar{\hat{r}}$ denote the i-th predicted reward and the batch mean. To prevent gradient flow through the correlation term, we introduce r_{sg} as a stop-gradient copy of r, treating it as a constant in the computation. The Pearson loss $\mathcal{L}_{pearson}$ is designed to maximize the correlation between $\hat{r}$ and r_{sg}, thereby driving the fitting model to accurately capture the reward model's length bias. In this formulation, the Pearson correlation coefficient is just employed as an optimization objective to capture the dependency between the two rewards, without implying assumption of linearity in length-reward relation. The fitting model itself is responsible for learning the underlying non-linear relation. In other words, the Pearson term is used only to align the correlation pattern between the two rewards, while the nonlinear fitting model independently learns the actual reward-length dependency.

3.3 Length Debiasing

After fitting the length bias through the fitting model, we now debias the reward model that was initially trained in warm-up stage by incorporating two critical objectives into its training: (1) preserving its discriminative capacity for human preferences, and (2) decoupling its outputs from response length dependence. This is achieved through a composite loss function:

$$\mathcal{L}_{debiased} = \mathcal{L}'_{pearson} + \mathcal{L}_{BT}. \quad (5)$$

Compared to $\mathcal{L}_{pearson}$, $\mathcal{L}'_{pearson}$ is slightly adapted to ensure gradient backpropagation through the reward model:

$$\mathcal{L}'_{pearson} = |\rho(r, \hat{r}_{sg})|. \quad (6)$$

The $\mathcal{L}'_{pearson}$ is to make the output of the reward model as uncorrelated as possible to the predicted reward of the fitting model, and the $\mathcal{L}_{BT}$ is to ensure that the model still has the ability to model human preferences. Additionally, to better fit the bias of the model, these two models take turns to train and the loss function could be written as:

$$\mathcal{L} = I(step) * \mathcal{L}_{debiased} + (1 - I(step)) * \mathcal{L}_{fit}. \quad (7)$$

Here $I(step)$ is the indicator function that indicates which model is trained under this step. For example, if we use every a (in our training a equals to 8) steps to change the model for training in the third stage, then $I(step)$ can be expressed as:

$$I(step) = \begin{cases} 0, \ 2ka \leq step < 2ka + a, \ k \in \mathbb{N}, \\ 1, \ 2ka + a \leq step < 2(k+1)a, \ k \in \mathbb{N}. \end{cases} \quad (8)$$

4 Experiments

In this section, we introduce the experimental settings and validate the effectiveness of our method through three key steps. First, we train the reward model to demonstrate its accuracy under different subsets and plot the length-reward distribution. Next, we apply reward models to various alignment algorithms to verify its effectiveness. Finally, we conduct an analysis of the length distribution between different methods and show the fitted curve of $model_f$ at different steps in training.

4.1 Experimental Settings

Training Data For training data, we utilize the static split Dahoas-rm-static[1] from Anthropic's HH dataset (Bai et al., 2022), partitioning it into three subsets: $30k$ samples for supervised fine-tuning (SFT) of the base model, $30k$ samples for reward model training, and $8k$ samples reserved for downstream task validation. Moreover, the dataset also contains $5k$ samples for testing (in Table 1).

Training Details In our experiments, we utilize the Qwen2.5-7B[2] and Gemma2-9B[3] models (Qwen et al., 2025; Team et al., 2024b), training them with the DeepSpeed framework (Rasley et al., 2020). For supervised fine-tuning, we employ a learning rate of 1e-5 with a batch size of 8 (per-device batch size 8) and training model for 2 epochs. Furthermore, all reward models are initialized from the same SFT model and trained use the learning rate of 2e-5, a batch size of 16 (per-device batch size 4 with gradient accumulation steps 4), and runs 1 epoch. We conduct these experiments on same hardware configurations: all models are trained on 8×A100 GPUs. Additionally, inference operations leverage vLLM (Kwon et al., 2023), with sampling configured as temperature 0.7, top-p 0.8, top-k 20, and repetition penalty 1.1.

Alignment Algorithm Given the computational demands and hyperparameter sensitivity of PPO, we focus on the evaluation of the BoN (Best of N) (Gui et al., 2024; Sessa et al., 2025; Dong et al., 2023) and DPO (Direct Preference Optimization) (Rafailov et al., 2023) approaches. The BoN implementation selects highest-scoring responses from N seed-generated outputs (here we set $N = 8$). In DPO, reward models cannot be directly applied, we utilize the original $8k$ preference pairs from the dataset but replace the human labels with predictions from reward models: both responses are scored, and the preference label is reassigned according to the score. All methods share same DPO training hyperparameters: learning rate 2e-6, 1 epoch, batch size 16 (per-device batch size 2 with gradient accumulation steps 8), and $\beta = 0.1$.

Compared Methods We compare three methods: the vanilla reward model (Vanilla RM) and two length-debiasing approaches based on a linear assumption, Length Penalty (Singhal et al., 2023) and ODIN (Chen et al., 2024b). The Length Penalty directly subtracts the product of response length and a fixed coefficient from the reward, whereas ODIN employs a two-head structure that linearly disentangles the reward model's assessments of length and quality, thereby mitigating length bias.

Evaluation To address the well-documented length bias in LLM evaluation (observed even in models like GPT (OpenAI et al., 2024)), we employ the length-controlled Alpaca-Eval (Dubois et al., 2024) benchmark for length debiased performance assessment. The key metrics include:

- Win Rate (WR): The winning rate of the aligned language model against fixed reference outputs (here we take the SFT model outputs as references because all methods share the same SFT model), as scored by GPT-4.
- Length-Controlled Win Rate (LC-WR): The win rate after applying length debiasing to GPT-4's evaluation. It is a more accurate metric that reflects the text generation quality.
- Length of characters/tokens (L_{char}/L_{token}): Average response length in characters/tokens.

[1] https://huggingface.co/datasets/Dahoas/rm-static

[2] https://huggingface.co/Qwen/Qwen2.5-7B

[3] https://huggingface.co/google/gemma-2-9b

Apart from Alpaca-Eval, we test the models trained with DPO on other benchmarks: MT-Bench (Zheng et al., 2023) and IFEval (Zhou et al., 2023). MT-Bench is a benchmark designed to evaluate the capabilities of LLMs, particularly focusing on their performance in multi-turn conversations and instruction-following tasks, and IF-Eval specifically designed to evaluate the instruction-following capability of LLMs.

4.2 Experimental Results

Accuracy on Preference Datasets We evaluate the accuracy of our method in preference datasets by splitting the test set into two subsets: C-longer containing samples where the chosen response was longer than the rejected response ($len(y_w) > len(y_l)$) and R-longer is the rejected response was longer ($len(y_l) > len(y_w)$). The results in Table 1 demonstrate that our approach achieves better length balance, with the reward model showing a closer accuracy across two subsets.

As shown in Table 1, FiMi-RM reduces accuracy on the C-longer subset while improving it on R-longer. We clarify that this trade-off is not a degradation of discriminative ability. A reward model with length bias trivially achieves high C-longer accuracy by exploiting the length shortcut. In the extreme case, ranking purely by length would yield 100% C-longer accuracy with no semantic understanding. Furthermore, since the C-longer subset constitutes a larger portion (58%) of the total dataset, a length-biased model can inflate its overall accuracy simply by optimizing on this majority subset. So the drop in C-longer accuracy therefore indicates that FiMi-RM has successfully discarded this shortcut, and the slight reduction in overall accuracy reflects a more balanced performance distribution rather than diminished capability. Importantly, the consistent improvements across downstream benchmarks confirm that semantic quality is preserved and improved after debiasing, demonstrating that the trade-off in static preference accuracy does not hinder the performance.

Length-Reward Distribution To further analyze the test set results, we plot a scatter graph of response length [4] and reward given by different 7B models, along with the average reward for different length ranges. Since directly calculating the mean reward for each individual length results in high variance, we split the length range into bins of size 25 to compute the average reward within each bin. As shown in Figure 2, our method exhibits a more balanced distribution compared to other methods: Our scatter plot demonstrates better symmetry along the x-axis (length axis), and average curve is also more flatter. This figure further validate the effectiveness of our length debiasing approach.

Performance of Different Alignment Algorithms Using Reward Models As shown in Table 2 and Table 3, we conduct an evaluation under two alignment algorithms: DPO and BoN. Our results reveal that the reward model consistently achieves higher LC-WR compared to baseline methods, demonstrating its effectiveness in mitigating length bias. While non-debiased win rate also favors ours.

In terms of response length, our method exhibit shorter outputs compare to others in most case, though our BoN-generated responses are longer than ODIN's in Qwen2.5-7B. However, shorter length does not always indicate better performance, the better length-controlled win rate confirms that our model produces optimally balanced responses within a reasonable length range.

To further evaluate performance, we assess the language model trained with DPO on additional benchmarks (Table 4). On MT-bench, our model outperforms competing methods in the first round (T1), second round (T2), and overall average score (Avg). It also achieves higher prompt-level accuracy on IFEval, offering further evidence that our length debiasing improves output quality.

Length Distribution of Reward Models' Selection We analyze the length distribution of responses selected by different 7B reward models under the BoN algorithm, which is shown in Figure 3 and Figure 4, as well as the distribution of chosen (y_w) and rejected (y_l) responses during DPO data annotation, which is shown in Figure 5 and 6.

The results in Figure 3 and 4 show that, compared to vanilla RM and Length Penalty, our method exhibits stronger preferences for shorter responses in BoN selection. While ODIN also reduces bias toward excessively long outputs, it primarily shifts preferences toward medium-length responses rather than increasing selection of shorter ones. In contrast, our approach has a more balanced distribution, with a tendency to favor concise outputs.

Furthermore, in DPO annotation (Figure 5 and 6), the gap between chosen and rejected response

[4] Unless otherwise specified, all subsequent references to "length" in this paper refer to token length

RM Acc (%)	Qwen2.5-7B			Gemma2-9B		
	All	C-longer	R-longer	All	C-longer	R-longer
Vanilla RM	70.14	**80.72**	56.88	66.42	**80.86**	47.63
Length Penalty	**70.45**	79.76	59.01	67.31	79.41	51.81
ODIN	70.08	78.61	59.50	**67.81**	76.81	56.69
FiMi-RM (Ours)	69.75	73.60	**65.70**	67.34	69.07	**66.38**

Table 1: Accuracy on Preference Datasets. Our approach achieves better length balance, with the reward model showing a closer accuracy across both subsets. Notably, the C-longer subset constitutes a larger proportion (58%) of the total dataset. Given this data distribution, prioritizing accuracy optimization for the C-longer subset may result in a misleadingly favorable assessment of overall performance.

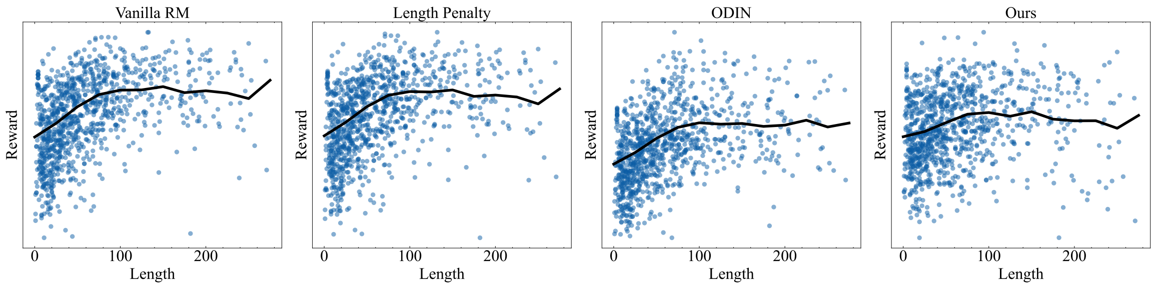

Figure 2: Scatter plot of reward-length, with binned averages (black lines). Our method demonstrates a more balanced reward distribution compared to others, indicating effective debiasing. Because the absolute reward magnitude is unconstrained during RM training, it is not comparable across different reward models; therefore, we do not label the y-axis in the figure.

BoN	Qwen2.5-7B				Gemma2-9B			
	LC-WR↑	WR↑	L_{char}	L_{token}	LC-WR↑	WR↑	L_{char}	L_{token}
Vanilla RM	68.25	73.84	638	148	62.91	65.77	756	179
Length Penalty	69.22	74.54	620	144	62.69	65.62	712	158
ODIN	71.38	75.58	512	115	63.46	65.24	684	168
FiMi-RM (Ours)	**72.59**	**76.39**	543	125	**66.68**	**67.77**	534	123

Table 2: The length-controlled Alpaca-Eval results under the BoN algorithm. Our method achieved the highest win rate (WR) and length-controlled win rate (LC-WR). While our output length in Qwen2.5-7B is slightly longer than ODIN's, we maintain better performance in LC-WR and WR, indicating more effective bias mitigation.

DPO	Qwen2.5-7B				Gemma2-9B			
	LC-WR↑	WR↑	L_{char}	L_{token}	LC-WR↑	WR↑	L_{char}	L_{token}
Vanilla RM	58.56	61.73	1089	254	57.35	59.81	1135	267
Length Penalty	57.67	62.89	980	227	57.78	60.03	1077	250
ODIN	58.91	63.16	1044	244	57.96	61.19	905	214
FiMi-RM (Ours)	**62.17**	**67.32**	757	174	**59.47**	**61.61**	773	189

Table 3: The length-controlled Alpaca-Eval results under the DPO algorithm. Our results demonstrate better performance in both LC-WR and WR.

length distribution is smaller for our method compared to other methods. This indicates that our reward model has less length bias in preference data labeling. The reduced discrepancy between chosen and rejected length further validates that our strategy mitigates length reward hacking better.

Training Process of Fitting Model In Figure 7 we show the fitted curve of $model_f$ at different steps in training stage 2 (stage of length bias fitting). The blue scatter points represent the actual

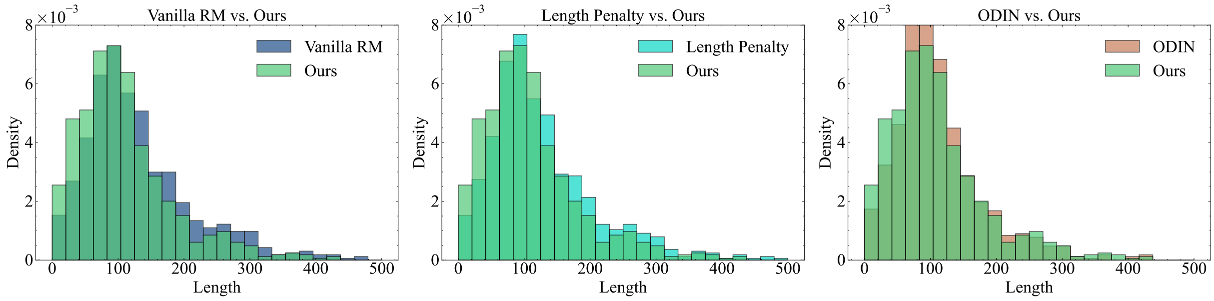

Figure 3: The pairwise comparison of the distribution of responses selected by BoN. The figure indicates that, relative to vanilla RM and Length Penalty, our approach demonstrates a stronger inclination toward shorter responses in BoN selection. Although ODIN also mitigates bias toward overly lengthy outputs, it mainly shifts preferences toward medium-length responses rather than enhancing the selection of shorter ones.

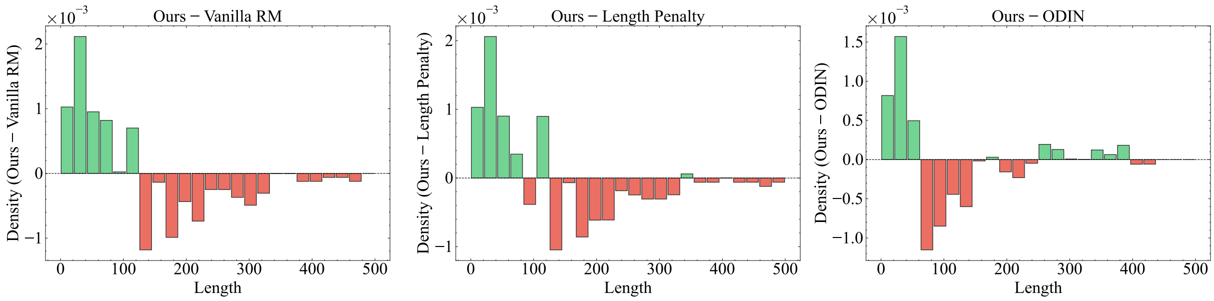

Figure 4: Difference in response length distributions under BoN, computed from the two overlapping histograms shown in Figure 3. Positive values (green) indicate that FiMi-RM assigns higher density to the corresponding length range compared to the others; negative values (red) indicate lower density.

DPO	**Qwen2.5-7B**				**Gemma2-9B**			
	MT-bench			IFEval	MT-bench			IFEval
	T1↑	T2↑	Avg↑	Acc_{prompt} ↑	T1↑	T2↑	Avg↑	Acc_{prompt} ↑
Vanilla RM	5.45	3.43	4.44	17.4	3.99	3.06	3.53	18.0
Length Penalty	5.11	3.35	4.23	16.6	4.51	3.18	3.84	18.2
ODIN	5.33	3.73	4.53	18.6	4.46	3.16	3.81	17.8
FiMi-RM (Ours)	**5.60**	**4.04**	**4.82**	**19.0**	**4.65**	**3.34**	**3.99**	**19.6**

Table 4: Other benchmarks' results under the DPO algorithm. Our model outperforms other methods on MT-bench in both the first round (T1), the second round (T2), and the average score. It also achieves improvements on prompt-level accuracy in IFEval, further demonstrating that our method enhances output quality after length debiasing.

output of the reward model before debiasing, while the black curves illustrate the relation fitted by the fitting model. Here we show the process of the fitting model capturing the bias relation and the fitted result. As shown in the figure, the initially fitted curve (step 0) exhibits no clear pattern at the beginning because it is randomly initialized, and with training progresses, the curve gradually aligns with the trend of the scatter points.

Furthermore, the last subfigure (step 200) reveals two distinct phases: For responses shorter than 100 tokens, the reward-length relation exhibits a strong linear trend, indicating that the bias grows almost proportionally with length. When beyond 100 tokens, the relation becomes noticeably flatter and in some cases even shows a slight downward tendency, suggesting that the effect of length on reward diminishes or reverses for longer outputs.

5 Conclusion

This paper primarily investigates length debiasing in reward models. Previous approaches to length debiasing are typically not characterize the bias form or assume a linear relation between input length and the hacked reward. To achieve better length debiasing, we employ a lightweight model

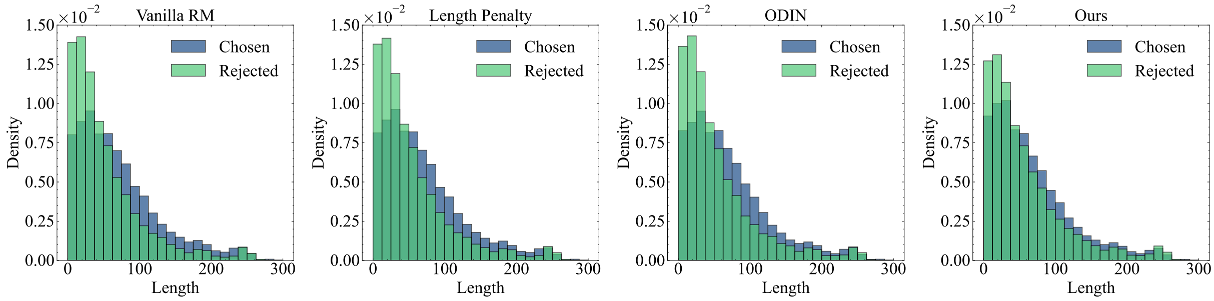

Figure 5: Length distribution of chosen and rejected responses in the labeling stage of DPO. The gap between chosen and rejected response length is noticeably smaller for our method when comparing to others.

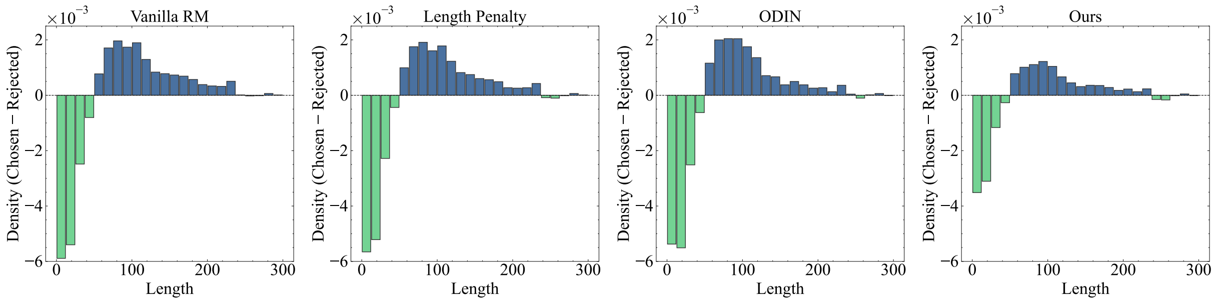

Figure 6: The difference between the chosen and rejected response length distributions (chosen - rejected) shown in Figure 5.

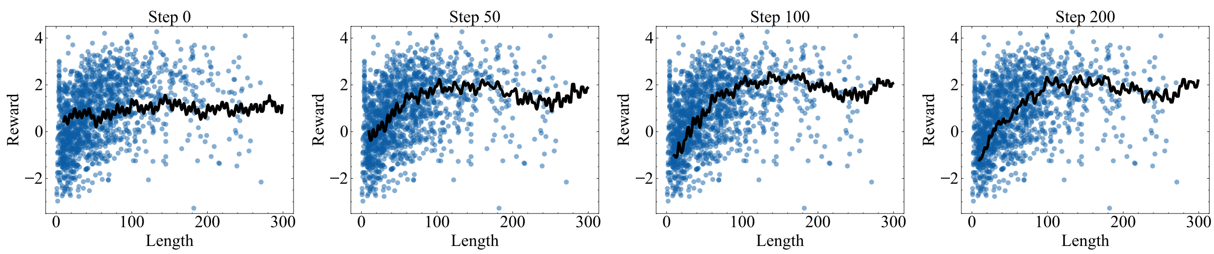

Figure 7: The fitted curve of $model_f$ at different steps in training.

to explicitly fit the relation between input length and the hacked reward from the reward model. In experiments, we test our method on BoN and DPO, observing improvements in different benchmarks. From the perspective of societal impact, better aligning models with human preferences helps them serve humanity more effectively and safely. However, stronger alignment capabilities could also be misused to align with harmful content. Therefore, we must strengthen the regulatory.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Contract 623B2097, the Youth Innovation Promotion Association CAS. It was also supported by the GPU cluster built by MCC Lab of USTC & the Supercomputing Center of USTC.

Limitations

Here we focus on length debiasing in reward models. Although we have achieved better results in debiasing, whether human preferences are entirely independent of length (or in terms of correlation, whether the Pearson correlation coefficient is truly zero) remains a question worthy of further investigation. From a practical standpoint, empirical observations suggest that humans often favor more detailed responses, which naturally tend to be longer. For instance, in tasks like summarization or open-ended question answering, thorough explanations with supporting evidence are typically rated higher than brief, vague answers; or to put it another way, sometimes users explicitly include requests for longer and more detailed responses in their instructions. These introduce a potential positive correlation between length and preferences.

References

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, and 3 others. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *Preprint*, arXiv:2204.05862.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Yuyan Bu, Liangyu Huo, Yi Jing, and Qing Yang. 2025. Beyond excess and deficiency: Adaptive length bias mitigation in reward models for RLHF. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3091–3098, Albuquerque, New Mexico. Association for Computational Linguistics.

Huayu Chen, Guande He, Lifan Yuan, Ganqu Cui, Hang Su, and Jun Zhu. 2024a. Noise contrastive alignment of language models with explicit rewards. *Preprint*, arXiv:2402.05369.

Lichang Chen, Chen Zhu, Jiuhai Chen, Davit Soselia, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. 2024b. ODIN: Disentangled reward mitigates hacking in RLHF. In *Forty-first International Conference on Machine Learning*.

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. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025b. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

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

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

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alexander Nicholas D'Amour, Krishnamurthy Dj Dvijotham, Adam Fisch, Katherine A Heller, Stephen Robert Pfohl, Deepak Ramachandran, Peter Shaw, and Jonathan Berant. 2024. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. In *First Conference on Language Modeling*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10835–10866. PMLR.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 4447–4455. PMLR.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Lin Gui, Cristina Gârbacea, and Victor Veitch. 2024. Bonbon alignment for large language models and the sweetness of best-of-n sampling. *arXiv preprint arXiv:2406.00832*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. ORPO: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189, Miami, Florida, USA. Association for Computational Linguistics.

Zeyu Huang, Zihan Qiu, Zili Wang, Edoardo Ponti, and Ivan Titov. 2025. Post-hoc reward calibration: A case study on length bias. In *The Thirteenth International Conference on Learning Representations*.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. Openassistant conversations - democratizing large language model alignment. In *Advances in Neural Information Processing Systems*, volume 36, pages 47669–47681. Curran Associates, Inc.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Nathan Lambert and Roberto Calandra. 2023. The alignment ceiling: Objective mismatch in reinforcement learning from human feedback. *arXiv preprint arXiv:2311.00168*.

Tianqi Liu, Wei Xiong, Jie Ren, Lichang Chen, Junru Wu, Rishabh Joshi, Yang Gao, Jiaming Shen, Zhen Qin, Tianhe Yu, Daniel Sohn, Anastasia Makarova, Jeremiah Zhe Liu, Yuan Liu, Bilal Piot, Abe Ittycheriah, Aviral Kumar, and Mohammad Saleh. 2025. RRM: Robust reward model training mitigates reward hacking. In *The Thirteenth International Conference on Learning Representations*.

Junru Lu, Jiazheng Li, Siyu An, Meng Zhao, Yulan He, Di Yin, and Xing Sun. 2024. Eliminating biased length reliance of direct preference optimization via down-sampled KL divergence. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1047–1067, Miami, Florida, USA. Association for Computational Linguistics.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems*, volume 37, pages 124198–124235. Curran Associates, Inc.

OpenAI. 2025. Expanding on what we missed with sycophancy.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 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. *Advances in neural information processing systems*, 35:27730–27744.

Richard Yuanzhe Pang, Vishakh Padmakumar, Thibault Sellam, Ankur Parikh, and He He. 2023. Reward gaming in conditional text generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4746–4763.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4998–5017, Bangkok, Thailand. Association for Computational Linguistics.

Karl Pearson. 1895. Notes on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

Alexandre Ramé, Johan Ferret, Nino Vieillard, Robert Dadashi, Léonard Hussenot, Pierre-Louis Cedoz, Pier Giuseppe Sessa, Sertan Girgin, Arthur Douillard, and Olivier Bachem. 2024a. Warp: On the benefits of weight averaged rewarded policies. *Preprint*, arXiv:2406.16768.

Alexandre Ramé, Nino Vieillard, Léonard Hussenot, Robert Dadashi, Geoffrey Cideron, Olivier Bachem, and Johan Ferret. 2024b. Warm: On the benefits of weight averaged reward models. *Preprint*, arXiv:2401.12187.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Pierre Harvey Richemond, Yunhao Tang, Daniel Guo, Daniele Calandriello, Mohammad Gheshlaghi Azar, Rafael Rafailov, Bernardo Avila Pires, Eugene

Tarassov, Lucas Spangher, Will Ellsworth, Aliaksei Severyn, Jonathan Mallinson, Lior Shani, Gil Shamir, Rishabh Joshi, Tianqi Liu, Remi Munos, and Bilal Piot. 2024. Offline regularised reinforcement learning for large language models alignment. *Preprint*, arXiv:2405.19107.

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

Pier Giuseppe Sessa, Robert Dadashi-Tazehozi, Leonard Hussenot, Johan Ferret, Nino Vieillard, Alexandre Rame, Bobak Shahriari, Sarah Perrin, Abram L. Friesen, Geoffrey Cideron, Sertan Girgin, Piotr Stanczyk, Andrea Michi, Danila Sinopalnikov, Sabela Ramos Garea, Amélie Héliou, Aliaksei Severyn, Matthew Hoffman, Nikola Momchev, and Olivier Bachem. 2025. BOND: Aligning LLMs with best-of-n distillation. In *The Thirteenth International Conference on Learning Representations*.

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. *arXiv preprint arXiv:2402.03300*.

Wei Shen, Rui Zheng, Wenyu Zhan, Jun Zhao, Shihan Dou, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. Loose lips sink ships: Mitigating length bias in reinforcement learning from human feedback. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2023. A long way to go: Investigating length correlations in rlhf. *arXiv preprint arXiv:2310.03716*.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, and 1118 others. 2024a. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024b. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Lilian Weng. 2024. Reward hacking in reinforcement learning.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Xuanchang Zhang, Wei Xiong, Lichang Chen, Tianyi Zhou, Heng Huang, and Tong Zhang. 2024. From lists to emojis: How format bias affects model alignment. *Preprint*, arXiv:2409.11704.

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. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 46595–46623.

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. *CoRR*, abs/2311.07911.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences. *Preprint*, arXiv:1909.08593.

A Ablation Study

We conduct ablation experiments with three dimensions: (1) the multi-stage training design vs. joint training, (2) the alternating training schedule in Stage 3, (3) the architecture of the fitting model $model_f$. All experiments use Qwen2.5-7B as the base model and BoN ($N = 8$) as the alignment algorithm, evaluated on AlpacaEval.

Multi-Stage Design vs. Joint Training. Here we explored a joint training strategy where the fitting model, reward model, and debiasing objective are optimized together. While joint training is feasible, its performance is weaker than the multi-stage design, as shown in Table 5. By separating bias modeling (Stage 2) and bias mitigation (Stage 3) into distinct stages, the pipeline allows the fitting model to first accurately characterize the length bias before the reward model attempts to remove it.

Alternating Training Schedule We compare FiMi-RM against a variant that trains only the reward model in Stage 3 (without alternating updates to $model_f$). As shown in Table 5, removing alternating training degrades both LC-WR and WR, confirming that refining the two models is beneficial for accurate bias capture and removal.

Architecture of Fitting Model We compare our ResNet-based $model_f$ against four alternatives: a simple linear function $f(x) = ax + b$, Polynomial Regression, 1D-CNN, and MLP. Results are summarized in Table 5. The ResNet architecture achieves the best performance.

B Visualization Results of Gemma Model

In the experimental section, we present various visualization results on the Qwen model, and in this appendix, we further provide corresponding results on the Gemma2-9B. Including: the scatter plot of reward-length (Figure 8); the comparison of the distribution of responses selected by BoN (Figure 9 and 10); the length distribution of chosen and rejected responses in the data reannotating stage of DPO (Figure 11 and 12).

From these figures, we observe that although the model family changes, the conclusions remain largely consistent with those obtained earlier on Qwen2.5-7B: our reward model exhibits a smaller bias with respect to length, demonstrating the generality of our approach.

C Detailed Network Structure of Fitting Model

Here we present the detailed structure of our $model_f$ in Table 6. As mentioned in method section, the formula for Length Encoding is

$$\mathrm{LE}(\mathrm{len}(y)) = \left[\sin\left(\frac{\mathrm{len}(y)}{10000^{2j/d}}\right), \cos\left(\frac{\mathrm{len}(y)}{10000^{2j/d}}\right) \right]_{j=0}^{\frac{d}{2}-1}. \tag{9}$$

Where the hidden dimension d is set to 32 in experiments. This yields a total of ≈ 6.4K parameters for $model_f$, which is negligible compared to the reward models' sizes (7B and 9B). Since the computation of the Pearson correlation coefficient becomes more accurate with larger batch size B, we adopt a multi-GPU aggregation framework, aggregating the batch data across 8 devices to obtain the length-reward pairs, this effectively increase the value of B. Additionally, due to the relatively small model size, we set the learning rate to 5e-3 during training.

D Computational Cost Analysis

FiMi-RM introduces a secondary fitting model, but this does not increase total training cost. To ensure a fair comparison, samples used to update the fitting model are not reused to update the reward model within the same iteration; consequently, the total number of training steps is identical to the Vanilla RM baseline.

Table 7 reports average per-step times on $8\times$A100 GPUs, across all 938 steps, FiMi-RM allocates 480 steps to first stage, 240 steps to Stage 2, and the remaining 218 steps to Stage 3. Since all steps in Stage 2 and half of the steps in Stage 3 are allocated to the faster fitting-model updates, the overall training time per epoch is reduced by approximately 25% relative to the Vanilla RM. FiMi-RM therefore achieves length debiasing with no additional computational overhead.

E BT Loss Monitoring During Stage 3

To assess whether the preference modeling capacity of the reward model is maintained during Stage 3, we monitor the Bradley–Terry loss $\mathcal{L}_{BT}$ throughout debiasing training. On the Dahoas-rm-static dataset with Qwen2.5-7B, the BT loss increases from 0.530 to 0.566 over Stage 3. This rise is an expected consequence of removing the length shortcut: because approximately 58% of chosen responses are longer, the vanilla model exploits

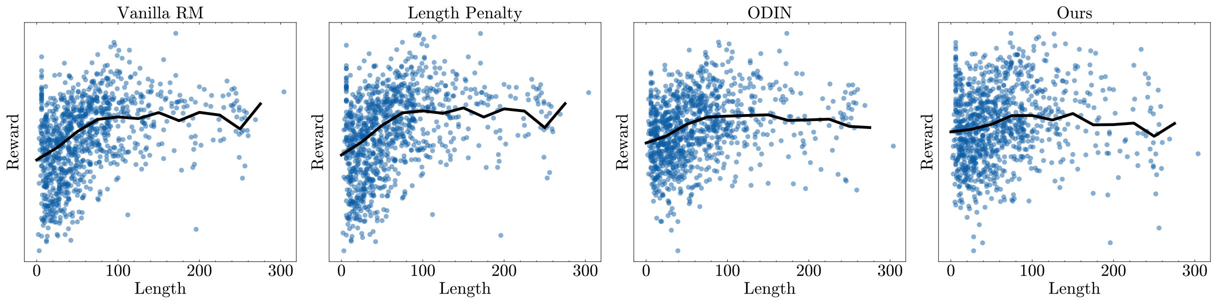

Figure 8: The scatter plot of reward-length relation (Gemma2-9B).

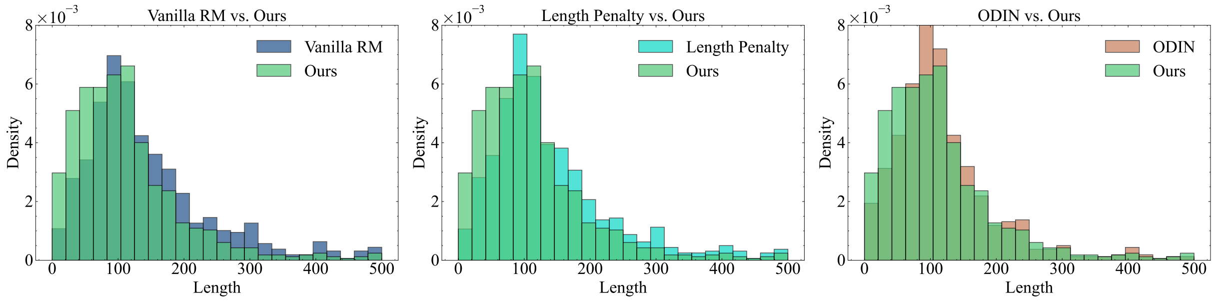

Figure 9: Comparison of the distribution of responses selected by BoN (Gemma2-9B).

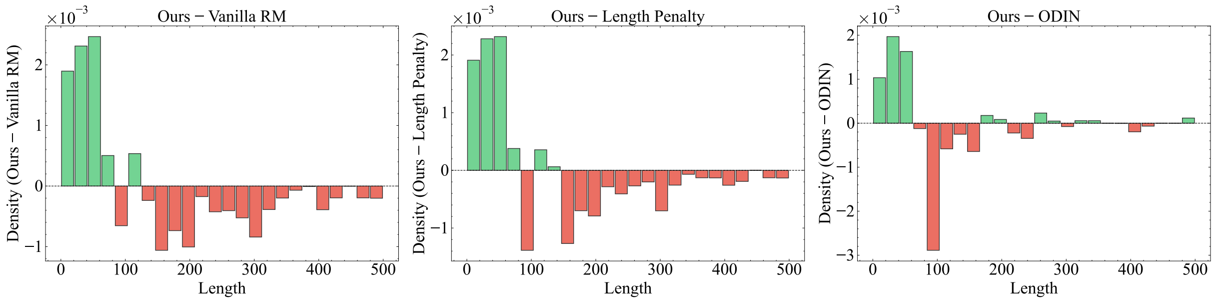

Figure 10: Difference in response length distributions (Gemma2-9B).

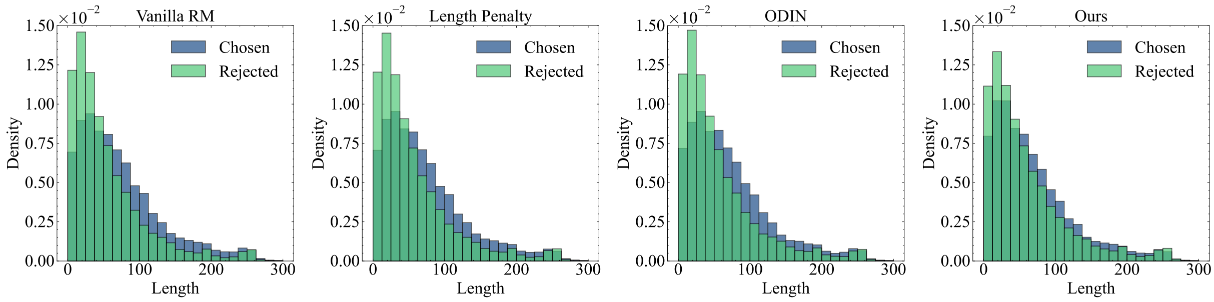

Figure 11: The length distribution of chosen and rejected responses in the labeling stage of DPO (Gemma2-9B).

length as a proxy signal, yielding an artificially low BT loss. By enforcing semantic focus and decoupling length from reward, the training objective becomes harder, which naturally raises the loss. Critically, consistent improvements across downstream benchmarks confirm that preference modeling ability is preserved and indeed strengthened after debiasing.

F Results under SimPO

To assess the generality of FiMi-RM beyond DPO and BoN, we conduct the experiments replacing DPO with SimPO (Meng et al., 2024) (learning rate 5e-7, $\beta = 2$, $\gamma = 0.8$, per-device batch size 2 and gradient accumulation steps 8 on Qwen2.5-7B). As shown in Table 8, FiMi-RM reduces response length and improves LC-WR compared to

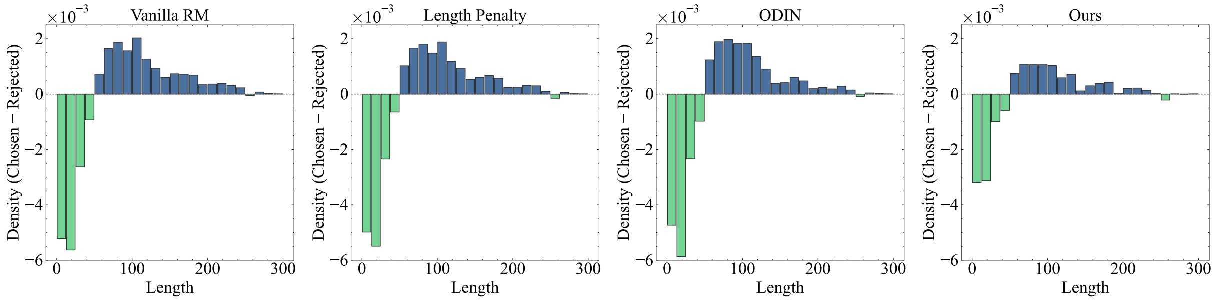

Figure 12: The difference between the chosen and rejected response length distributions (chosen - rejected) shown in Figure 11 (Gemma2-9B).

Method	LC-WR↑	WR↑	L_{char}
Vanilla RM	69.25	73.84	638
FiMi-RM with Joint Training	71.51	76.01	568
FiMi-RM without Alternating Training	71.16	74.27	588
FiMi-RM with Polynomial Regression	67.11	71.54	722
FiMi-RM with Linear Fitting	71.67	75.30	501
FiMi-RM with 1D-CNN	71.15	74.29	516
FiMi-RM with MLP	71.35	75.26	550
FiMi-RM (Ours)	**72.59**	**76.39**	543

Table 5: Ablation results. "Joint Training" optimizes all objectives together in a single stage. "without Alternating Training" trains only the reward model in Stage 3. "Linear Fitting" replaces $model_f$ with $f(x) = ax + b$.

component	layer	input → output
embedding	Length Encoding	$len(y)$:[B] → [B, d]
block 1	LayerNorm(d) MLP(d→d→d→d) ReLU + residual	[B, d] → [B, d] [B, d] → [B, d] [B, d] → [B, d]
block 2	LayerNorm(d) MLP(d→d→d→d) ReLU + residual	[B, d] → [B, d] [B, d] → [B, d] [B, d] → [B, d]
head	Linear(d→1) squeeze(1)	[B, d] → [B, 1] [B, 1] → $\hat{r}$:[B]

Table 6: Structure of $model_f$

the vanilla reward model under SimPO training as well.

G Evaluation on RLHFlow Dataset

To evaluate robustness across diverse preference distributions, we train FiMi-RM on the RLHFlow (Dong et al., 2024) dataset, a comprehensive dataset with eight sources. Table 9 reports AlpacaEval results using Qwen2.5-7B and BoN ($N = 8$).

FiMi-RM consistently outperforms the vanilla reward model on this more diverse dataset, achieving higher alignment accuracy with shorter outputs.

H Evaluation on other RLHF Setting

We futher evaluate our debiased reward model directly in other RLHF setting using GRPO (Shao et al., 2024). Table 10 reports AlpacaEval results on Qwen2.5-7B, with learning rate 2e-6, batch size 8 (per-device batch size 1 with gradient accumulation steps 8), and sample size $n = 8$. FiMi-RM achieves higher LC-WR while producing more concise responses.

I Use Of AI Assistants

Alpaca-Eval and MT-Bench use GPT for evaluation; aside from that, AI is used only for polishing and grammar checks.

Training Step Type	Time (s/step)
Standard RM update (Vanilla RM baseline)	$\approx$2.48
Fitting model update (forward through both; backward through $model_f$)	$\approx$0.60
Debiased RM update (forward through both; backward through RM)	$\approx$2.50
Vanilla RM total	$\approx$**43 min/epoch**
FiMi-RM total	$\approx$**32 min/epoch**

Table 7: Training time on $8\times$A100 GPUs (Qwen2.5-7B). The fitting model (6.4k parameters) is negligibly small relative to Qwen2.5-7B (7B parameters), so its backward is much more faster. The total time includes not only training time but also other overhead, so it will be longer than simply training time × number of steps.

SimPO	LC-WR↑	WR↑	L_{char}
Vanilla RM	53.68	58.69	711
FiMi-RM	**55.03**	**58.81**	549

Table 8: AlpacaEval results under SimPO training (Qwen2.5-7B).

Method	LC-WR↑	WR↑	L_{char}
Vanilla RM	62.00	68.73	1627
FiMi-RM (Ours)	**66.59**	**71.72**	1595

Table 9: AlpacaEval results on the RLHFlow dataset (Qwen2.5-7B, BoN).

GRPO	LC-WR↑	WR↑	L_{char}
Vanilla RM	58.80	72.73	1285
FiMi-RM	**67.42**	**75.91**	999

Table 10: Results under GRPO training.