Prior Constraints-based Reward Model Training for Aligning Large Language Models

Hang Zhou[1], **Chenglong Wang**[1], **Yimin Hu**[1], **Tong Xiao**[1,2]*,
Chunliang Zhang[1], **and Jingbo Zhu**[1,2]
[1]NLP Lab, School of Computer Science and Engineering,
Northeastern University, Shenyang, China
[2]NiuTrans Research, Shenyang, China
{stceum, clwang1119}@gmail.com,
{xiaotong, zhujingbo}@mail.neu.edu.cn

Abstract

Reinforcement learning with human feedback for aligning large language models (LLMs) trains a reward model typically using ranking loss with comparison pairs. However, the training procedure suffers from an inherent problem: the uncontrolled scaling of reward scores during reinforcement learning due to the lack of constraints while training the reward model. This paper proposes a **P**rior **C**onstraints-based **R**eward **M**odel (**PCRM**) training method to mitigate this problem. PCRM incorporates prior constraints—specifically, length ratio and cosine similarity between outputs of each comparison pair—during reward model training to regulate optimization magnitude and control score margins. We comprehensively evaluate PCRM by examining its rank correlation with human preferences and its effectiveness in aligning LLMs via RL. Experimental results demonstrate that PCRM significantly improves alignment performance by effectively constraining reward score scaling. As another bonus, our method is easily integrated into arbitrary rank-based alignment methods, such as direct preference optimization, and can yield consistent improvement. The code is available at https://github.com/wangclnlp/DeepSpeed-Chat-Extension/tree/PCRM.

1 Introduction

Reinforcement learning with human feedback (RLHF) has been proven to be an advanced technology to align large language models (LLMs) with human preferences (Ouyang et al., 2022; Ji et al., 2023; Wang et al., 2023b). It builds upon preference data, which rates and compares different outputs given the same input, where this rating is conducted by either human annotators or LLMs (Ouyang et al., 2022; Lee et al., 2023; Cui et al., 2023; Dubois et al., 2024). In practice, RLHF trains a reward model on the preference data with ranking loss for higher scores on preferred outputs than dispreferred ones. Then, RL algorithms such as Proximal Policy Optimization (PPO) (Schulman et al., 2017) are employed to fine-tune the LLM with the aim of optimizing this reward. During RL training, the reward model will give scores as signals, and the LLM will align the preference by increasing the probabilities of sampled outputs with higher reward scores. It is discernible that the alignment of the LLM is significantly influenced by how well the reward model is trained.

However, the training procedure of the reward model actually suffers from a failure mode: it learns from the preference data with the standard ranking loss yet provides a reward score for the sampled outputs during RL training. This training mode causes an inherent problem: the ranking loss increases the score margin between the outputs of this comparison without constraint during the training procedure, which makes an uncontrollable scale of scores in the process of RL training. For instance, the reward model trained by ranking loss can predict reward scores with a relatively large margin to two sampled outputs that do not differ very much (Zhu et al., 2023).

To address the problem, we propose a **P**rior **C**onstraints-based **R**eward **M**odel (**PCRM**) training method in this paper, which incorporates prior constraints while training the reward model. Specifically,

*Corresponding author.

we select two features of the preference data for designing prior constraints: length ratio and cosine similarity of the outputs with the same prompt. The length ratio is computationally less demanding and directly reflects data differences, while the cosine similarity captures deeper semantic features despite being more computationally intensive. These constraints effectively regulate the optimization magnitude across different outputs for the same input, thereby controlling the margin of scores predicted by the reward model.

We comprehensively evaluate the proposed PCRM in the following two ways. Firstly, we test the rank correlation between the preferences predicted by PCRM and human preferences. It can assess the extent to which the reward model can serve as a surrogate for human-derived preference signals. Secondly, we verify the effectiveness of the PCRM in aligning an LLM via RL. This also demonstrates the influence of constraints on alignment performance straightly. Notably, PCRM can yield a +2.48% improvement in the GPT-4 win rate for the dialogue task compared to the traditional RLHF. Furthermore, we integrate our method into direct preference optimization (DPO), a rank-based alignment method. The results show that our method can also be effective in improving the rank-based alignment methods, *e.g.,* a 2.95% increase in the GPT-4 win rate on the dialogue task compared to DPO.

2 Related Work

Reinforcement learning with human feedback (RLHF) is a crucial technique to ensure that the behaviours of large language models (LLMs) are consistent with human preferences (Stiennon et al., 2020; Ouyang et al., 2022; Wang et al., 2023b). Recent works resorted to building a better reward model to improve the performance of RLHF on LLMs. These methods could be classified into three groups. The first group aimed to efficiently produce human preference data (Dubois et al., 2024; Cui et al., 2023; Lee et al., 2023). For instance, Lee et al. (2023) directly employed an LLM to annotate the comparison pairs. Significantly, the reward model trained on the LLM-annotated comparison pairs can achieve closer performance when applied to RLHF than the one trained on human-annotated comparison pairs. The second group tended to design fine-grained reward models to provide multiple reward scores for different reward criteria (Cheng et al., 2023; Wang et al., 2023b; Wu et al., 2024; Zhong et al., 2024). Notably, Wang et al. (2023b) learned various evaluation models from an LLM as reward models on the summarization task, including reward models scored for relevance, scored for fluency, scored for consistency, and scored for coherence. The third group that has attracted less attention generally explored how to merge multiple reward models in RLHF, such as reward model ensemble (Coste et al., 2023) and learning reward weights (Min et al., 2024). Different from these methods, in this paper, we proposed a superior training schema that enables this reward model to give a more accurate score in RLHF. Specifically, we refined the conventional ranking loss, typically utilized in reward model training, by incorporating prior constraints. These constraints are designed to regulate the margin between the scores of various outputs generated from the same input.

To the best of our knowledge, there is almost no previous work in training a reward model with prior constraints. The few related one is LLaMA2 (Touvron et al., 2023), which added a margin component in ranking loss for training each comparison pair. Although this margin simulates a constraint, it requires a manual annotation, which significantly increases the cost of training the reward model. Additionally, this work did not provide a specialized exploration or analysis on the topic of constraining the margin of scores during training reward models. In comparison, all of our prior constraints were computed automatically and did not require any human annotation. We also provided sufficient theoretical analysis and experiments to constrain the margin of scores. Furthermore, although there are some methods such as DPO (Amini et al., 2024) and RRHF (Yuan et al., 2023) that circumvent the need for training reward models, they still suffer from an inherent limitation from ranking loss, which is mentioned in DPO (Amini et al., 2024) without a solution. Our proposed PCRM can be easily extended to these methods to yield some benefits (See Section 5.7.2).

3 Background

Human-preference alignment training is a key technique to ensure that the behaviours of LLMs are consistent with human preferences. Recent efforts to align LLMs have mainly been conducted via RLHF. It typically includes three stages: 1) collecting preference data, 2) training a reward model with ranking loss, and 3) optimizing an LLM against the reward model via RL.

3.1 Collecting Preference Data

The preference data consists of the given input x and the corresponding different outputs sampled from LLMs trained by SFT (denoted as π_θ^{SFT}) or annotated by humans, which we refer to as $(y_1, y_2, \cdots, y_n|x)$. In the preference data, the different outputs are rated and ranked by humans or LLMs with specific aspects (Ouyang et al., 2022; Dubois et al., 2024), which are denoted as $(y^{(1)} \succ y^{(2)} \succ \cdots \succ y^{(i)} \succ y^{(j)} \succ \cdots \succ y^{(n)}|x)$, where $y^{(1)}$ is the best while $y^{(n)}$ is the worst.

3.2 Training Reward Model

After preference data collection, we can train a reward model π_θ^{RM} from the preference data, where the training objective is to fit human preference. Note that we use π_θ^{SFT} to initialize the reward model. Suppose r^* as the ideal model of human preference, based on the Bradley-Terry model (Bradley and Terry, 1952), the distribution of human preference $P_{\text{Human}}\left(y^{(i)} \succ y^{(j)}|x\right)$ can be written as:

$$P_{\text{Human}}\left(y^{(i)} \succ y^{(j)}|x\right) = \frac{\exp\left(r^*\left(y^{(i)}, x\right)\right)}{\exp\left(r^*(y^{(i)}, x)\right) + \exp\left(r^*(y^{(j)}, x)\right)} \tag{1}$$

$$= \sigma\left(r^*(y^{(i)}, x) - r^*(y^{(j)}, x)\right) \tag{2}$$

where $y^{(i)}$ denotes the data ranking i and σ denotes the Sigmoid activation function. When dealing with multiple outputs more than two, similarly, we can induce $P_{\text{Human}}(\cdot)$ based on the more general Plackett-Luce model (Plackett, 1975; Luce, 2005):

$$P_{\text{Human}}\left(y^{(1)} \succ y^{(2)} \succ \cdots \succ y^{(i)} \succ y^{(j)} \succ \cdots \succ y^{(n)}|x\right) = \prod_{i=1}^{n} \frac{\exp\left(r^*(y^{(i)}, x)\right)}{\sum_{j=i}^{n} \exp\left(r^*(y^{(j)}, x)\right)} \tag{3}$$

To learn the preference distribution, we need to increase the probability of preferred outputs. Here, it is typically achieved by a negative log-likelihood loss function:

$$\mathcal{L}_r = -\log P_{\text{Human}}\left(y^{(1)} \succ y^{(2)} \succ \cdots \succ y^{(i)} \succ y^{(j)} \succ \cdots \succ y^{(n)}|x\right) \tag{4}$$

$$= -\sum_{i=1}^{n} \log \frac{\exp\left(r^*\left(y^{(i)}, x\right)\right)}{\sum_{j=i}^{n} \exp\left(r^*\left(y^{(j)}, x\right)\right)} \tag{5}$$

Specially, when $n = 2$ which means there is only a comparison pair, the loss would be:

$$\mathcal{L}_r = -\log P_{\text{Human}}\left(y^{(i)} \succ y^{(j)}\right) \tag{6}$$

$$= -\log \sigma\left(r^*\left(y^{(i)}, x\right) - r^*\left(y^{(j)}, x\right)\right) \tag{7}$$

3.3 RL Training Against the Reward Model

In the process of RL training, we use the reward output π_θ^{RM} as signals, combined with an RL algorithm. Taking PPO as an instance, the corresponding loss for this training sample is given by:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta}\left[\pi_\theta^{\text{RM}}(x, y)\right] - \beta \mathbb{D}_{\text{KL}}\left[\pi_\theta || \pi_\theta^{\text{SFT}}\right] \tag{8}$$

where $\mathcal{D}$ is the dataset of RL training, x is the input, and y for the sampled outputs. $\mathbb{D}_{\text{KL}}$ is the KL dispersion which measures the distributional difference between π_θ^{SFT} and π_θ, multiplied by β which controls their distance, as the bigger β is, the more significant constraint is applied.

4 Prior Constraints-based Reward Model

Motivated by the challenges posed by the uncontrollable scale of scores in the reward model during the RL training process, our aim is to constrain the score margin between the outputs of this comparison when training the reward model. We propose the PCRM method to achieve this. Unlike conventional ranking loss, the proposed PCRM can constrain the maximum score margin between the outputs of each comparison with the length and cosine similarity features. In the following subsections, we will describe our PCRM in detail.

4.1 Optimization Objective of PCRM

Given the input x and the reward model π_θ^{RM}, the probability of $y_1 \succ y_2$ can be written as:

$$\begin{aligned} & P_{\pi_\theta^{\text{RM}}}\left(y_1 \succ y_2 | x\right) && (9)\\ = \; & P_{\pi_\theta^{\text{RM}}}(y_1 - y_2 \succ 0 | x) && (10)\\ = \; & \sigma\left(\pi_\theta^{\text{RM}}(y_1, x) - \pi_\theta^{\text{RM}}(y_2, x)\right) && (11)\\ = \; & \sigma\left(\Delta_{\pi_\theta^{\text{RM}}}(y_1, y_2, x)\right) && (12) \end{aligned}$$

where $\Delta_{\pi_\theta^{\text{RM}}}(\cdot)$ denotes the margin in the evaluation scores predicted by the reward model π_θ^{RM}. If the reward model learns this probability with a standard ranking loss, the reward score of the preferred output will increase; conversely, the reward score of the dispreferred output will decrease. This endeavour aims to maximize the margin between the score of the preferred and dispreferred outputs as much as possible. However, in RLHF, we do not just expect the reward model to be able to distinguish which output is more preferred, but also to be able to give information on *how much* more one input is preferred. Consequently, we conjecture that facilitating the reward model to learn an appropriate score margin across different outputs could improve the performance of RLHF.

We achieve this goal by adding a maximum margin constraint, denoted as $\Delta^*(\cdot)$, where it takes values in the range $(0, +\infty)$. We re-derive Equation 9 with the constraint $\Delta^*(\cdot)$ as follows:

$$\begin{aligned} & P_{\pi_\theta^{\text{RM}}}\left(\Delta^*(y_1, y_2, x) \succ y_1 - y_2 \succ 0 | x\right) && (13)\\ = \; & P_{\pi_\theta^{\text{RM}}}\left(\Delta^* > \Delta_{\pi_\theta^{\text{RM}}} > 0\right) && (14)\\ = \; & P_{\pi_\theta^{\text{RM}}}\left(\Delta^* > \Delta_{\pi_\theta^{\text{RM}}}\right) \times P_{\pi_\theta^{\text{RM}}}\left(\Delta_{\pi_\theta^{\text{RM}}} > 0\right) && (15)\\ = \; & \sigma\left(\Delta^* - \Delta_{\pi_\theta^{\text{RM}}}\right) \times \sigma\left(\Delta_{\pi_\theta^{\text{RM}}}\right) && (16) \end{aligned}$$

Based on the above derivation, we have the negative log-likelihood loss similar to the vanilla one:

$$\begin{aligned} \mathcal{L}_{\text{PCRM}} &= -\log P_{\pi_\theta^{\text{RM}}}\left(\Delta^* > \Delta_{\pi_\theta^{\text{RM}}} > 0\right) && (17)\\ &= -\log \sigma\left(\Delta^* - \Delta_{\pi_\theta^{\text{RM}}}\right) - \log \sigma\left(\Delta_{\pi_\theta^{\text{RM}}}\right) && (18) \end{aligned}$$

We define the maximum margin constraint through a negative correlation with the similarity between y_1 and y_2:

$$\Delta^*(y_1, y_2, x) = \frac{\beta_1}{\text{Sim}(y_1, y_2, x) + \beta_2} + \beta_3 \tag{19}$$

where $\text{Sim}(\cdot)$ denotes the similarity of the different outputs. β_1 controls the magnitude of the $\Delta^*(\cdot)$, β_2 controls the range of variation and β_3 controls the offset. We employ length ratio and cosine similarity to estimate $\text{Sim}(\cdot)$. When using the length ratio to estimate $\text{Sim}(\cdot)$, $\text{Sim}(\cdot)$ can given by:

$$\text{Sim}_{\text{len\_rat}}(y_1, y_2, x) = \frac{\min(\phi(y_1), \phi(y_2))}{\max(\phi(y_1), \phi(y_2))} \tag{20}$$

where $\phi(y)$ denotes the length of y. Note that we do not need the input x to be involved when employing the length ratio to estimate the $\text{Sim}(\cdot)$. When considering the cosine similarity as another estimation of the $\text{Sim}(\cdot)$, written as:

$$\text{Sim}_{\text{cos\_sim}}(y_1, y_2, x) \quad = \quad 1 - \frac{1}{\pi} \times \arccos\left(\frac{E(x, y_1) \cdot E(x, y_2)}{\max(||E(x, y_1)||_2, \epsilon) \cdot \max(||E(x, y_2)||_2, \epsilon)}\right) \tag{21}$$

where ϵ is a small value to avoid division by zero and $\arccos(\cdot)$ is the inverse of the cosine function. Inspired by the strong text encoding capability of the pre-trained model (Devlin et al., 2018; Xiao and Zhu, 2023), we employ a pre-trained model like BERT to compute $E(\cdot)$. There are other choices to define $\text{Sim}(\cdot)$ for specific tasks. For instance, we can use the ROUGE function (Lin, 2004) to define $\text{Sim}(\cdot)$ in the summarization task.

4.2 Analysis of the Optimization for PCRM

To better understand the constrained optimization for PCRM, we take the derivative of the loss function. The gradient of $\mathcal{L}_{\text{PCRM}}$ with respect of the parameters θ is:

$$\nabla_\theta \mathcal{L}_{\text{PCRM}} = \nabla_\theta \Delta_{\pi_\theta^{\text{RM}}} \times \left(\sigma\left(\Delta_{\pi_\theta^{\text{RM}}} - \Delta^*\right) - \sigma\left(-\Delta_{\pi_\theta^{\text{RM}}}\right)\right) \tag{22}$$

As a comparison, we also take the derivative of one for vanilla reward loss, which can be written as:

$$\nabla_\theta \mathcal{L}_{\text{RM}} = \nabla_\theta \Delta_{\pi_\theta^{\text{RM}}} \times \left(-\sigma\left(-\Delta_{\pi_\theta^{\text{RM}}}\right)\right) \tag{23}$$

Compare Equation 22 with 23, we can find their difference $\sigma\left(\Delta_{\pi_\theta^{\text{RM}}} - \Delta^*\right)$ which is always positive and will decrease the coefficient of $\nabla_\theta \Delta_{\pi_\theta^{\text{RM}}}$, constraining the optimization thereby.

When $\sigma\left(\Delta_{\pi_\theta^{\text{RM}}} - \Delta^*\right) - \sigma\left(-\Delta_{\pi_\theta^{\text{RM}}}\right) = 0$ in Equation 22, it implies that $\Delta_{\pi_\theta^{\text{RM}}} - \Delta^* = -\Delta_{\pi_\theta^{\text{RM}}}$ due to the monotonic increase of the Sigmoid activation function. Thus, we can deduce $\Delta_{\pi_\theta^{\text{RM}}} = \frac{\Delta^*}{2}$. Then we can get the conclusion that when $\Delta_{\pi_\theta^{\text{RM}}} < \frac{\Delta^*}{2}$, the coefficient of $\nabla_\theta \Delta_{\pi_\theta^{\text{RM}}}$ has the same sign with the one of the vanilla reward gradient, meaning the same optimization direction with the origin; in contrast, when $\Delta_{\pi_\theta^{\text{RM}}} > \frac{\Delta^*}{2}$, there will be opposite optimization direction decreasing the margin of reward scores. In this way, we can control the distance of scores of different outputs while optimizing the reward model.

5 Experiments

We evaluate the proposed PCRM in the following two ways. Firstly, we test the reward model trained by PCRM. Secondly, we analyze the performance of applying this trained reward model to RLHF. We conduct experiments on the commonly used generation tasks, including dialogue and summarization.

5.1 Datasets

The datasets used for each task are as follows:

- *Dialogue*: We employed AlpacaFarm (Dubois et al., 2024) dataset on dialogue task, which consists of 10K supervised fine-tuning split, 10K pairwise preference split, 20K unlabeled split for RL training, and 2K validation split based on 52k Alpaca data (Taori et al., 2023). For evaluation, we employed their evaluation set, which contains 805 instructions selected from a series of open-source datasets with real-world user interactions as reference instructions.

- *Summarization*: We used the filtered versions of the TL;DR dataset and human feedback dataset provided by OpenAI (Stiennon et al., 2020) on summarization task, the former one for instruction tuning and alignment, and the latter one for reward modeling. The TL;DR dataset is filtered to ensure quality and contains 123.2K samples in the final version, including 116.7K for training, 6.4K for validating, and 6.5K for testing. The large, high-quality preference dataset of human comparisons between summaries contains a 92.9K training set, a 33.1K validation set, and a 50.7K test set. Due to the enormous computational cost caused by the vast testing set, we randomly selected 10% as our final test set.

5.2 Settings

Task	**Sim**$(\cdot)$	β_1	β_2	β_3	**max_len**
Dialogue	len_rat	10	0.001	-5	512
	cos_sim	20	0.001	-15	
Summarization	len_rat	10	0.001	-5	1024
	cos_sim	20	0.001	-15	

Table 1: The hyper-parameters used in our experiments.

We conducted our experiments based on DeepSpeed-Chat[1] with a cross-entropy loss on supervised fine-tuning, where prompt parts were masked, and a ranking loss was employed by reward modeling. We set the maximum sequence length to 512 and 1024 for the dialogue and summarization tasks, respectively. For calculating the similarity between pair-wise data for setting prior constraints, we employed the BERT-base model (Devlin et al., 2018) to encode paired data and then calculated the similarity between `[CLS]` embeddings. Considering that the maximum sequence length supported by BERT is 512; however, the sequence length is 1024 for summarization, we set the tokenizer to truncate from the left, *i.e.*, truncating part of the prompt and calculating the semantic similarity of the remaining part. Additionally, the value of β_1, β_2 and β_3 mentioned in Eq. 19 are shown in Table 1. The performance of other hyper-parameters is reported in Section 5.7.1. Furthermore, we employed the top-p sampling method in the process of generation, where the temperature and the p were set to 0.75 and 0.95, respectively.

5.3 Evaluation Metrics

We evaluated the effectiveness of our method comprehensively from two dimensions: training reward models and aligning LLMs with the trained reward models. For training reward models, we scored the pair-wise test set. Based on this, we calculated the accuracy of the predicted scores (for details of calculating accuracy, see Appendix A). We trained three epochs on the training set and saved the model for each epoch. We selected the best one with the validation set. We compared the performance of the vanilla method with ours, as shown in the following section. To align LLMs with the trained reward models, we used different metrics depending on the tasks. Specifically, for the dialogue task, we used PandaLM (Wang et al., 2023c) and GPT-4 to choose a better one from the model output and the reference and calculate the win rate following Rafailov et al. (2024). For the summarization task, except for GPT-4, we also employ ROUGE (Lin, 2004) and BARTScore (Yuan et al., 2021) to evaluate the quality of the summary generated by the model from the posts.

5.4 Results of Training Reward Models

The performance of the PCRM on the dialogue and the summarization task are presented in Table 2. We can see that the accuracy of the PCRM increases slightly, either by calculating similarity with cosine similarity or length ratio. As a comparison, we test the reward models trained with random and fixed constraints, whose performance is close to the vanilla one with some decrease. This result demonstrates the importance of the prior information. The appropriate value of the constraints also affects the performance because the training object of PCRM can be divided into two parts: achieving higher accuracy and controlling the distribution of the reward scores. The latter part may conflict with the former under extreme constraints (See Section 5.7.1).

To further explore the distribution of reward scores, we visualize the relationship between the margin of predicted reward scores and the similarity of the paired data on dialogue task, with or without constraints in Figure 1. The area with deeper colour in the figure represents more points gathering at that position. The points in Figure 1 (a) are more dispersed than the ones shown in Figure 1 (b) below, as the upper right part of the figures with constraints are more clean. From Figure 1 (b), we can observe that

[1] https://github.com/microsoft/DeepSpeedExamples

Method	Dialogue	Summarization
Vanilla RM	54.93	72.20
PCRM-Random	54.13	71.70
PCRM-Fixed	54.40	71.70
PCRM-Length (Ours)	55.87	72.50
PCRM-Cosine (Ours)	**56.53**	**72.60**

Table 2: The accuracy of reward models on the dialogue and summarization task. "-Random" means using random constraints, "-Fixed" means using fixed one, "-Cosine" means using constraints calculated by cosine similarity, and "-Length" for length ratio.

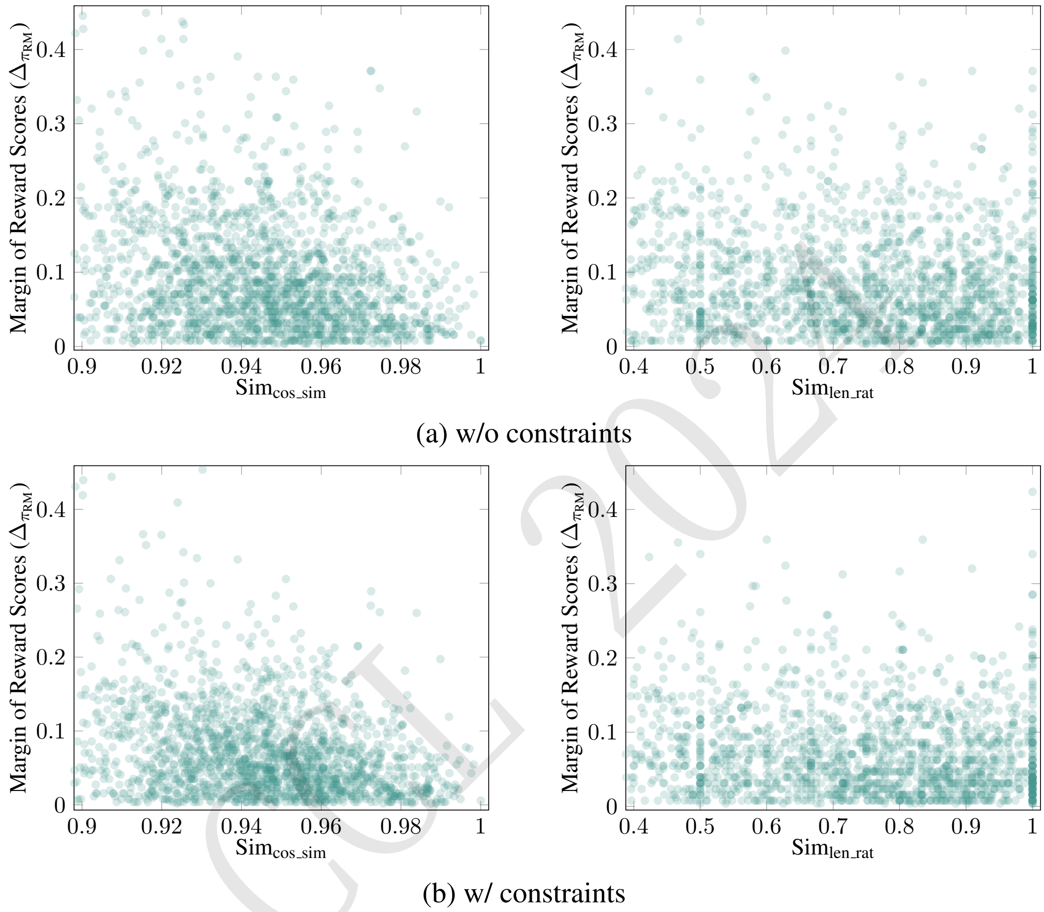


Figure 1: The distribution between the margin of predicted reward scores and the similarity of paired data (calculated by cosine similarity or length ratio) with or without constraint on the dialogue task. Each green point corresponds to a single data sample. The x-axis refers to the similarity calculated by the cosine similarity of sentence embedding or by the ratio of sentence length. The y-axis refers to the margin of the reward scores.

the margin of those points is limited to a lower level because of their higher similarity. This observation confirms that our method can provide an effective constraint during the training of the reward model. We can draw similar observations on the summarization task (See Figure 3 in Appendix B).

5.5 Results of Dialogue

The results of our alignment with PCRM on downstream tasks, as detailed in Table 3, show that PCRM significantly outperforms the vanilla reward model. Specifically, when utilizing cosine similarity, PCRM surpasses RLHF by +3.18 points on PandaLM and +2.48 points on the GPT-4 win rate in dialogue task.

Method	Dialogue		Summarization		
	PandaLM	GPT-4 Win	ROUGE-L	BARTScore	GPT-4 Win
SFT	65.75	55.77	22.93	-6.31	41.83
RLHF	72.73	60.13	25.10	-5.17	70.53
PCRM-Random	70.81	59.75	24.03	-6.02	50.53
PCRM-Fixed	69.34	60.71	24.87	-6.09	52.21
PCRM-Length (Ours)	73.52	62.30	22.76	**-4.90**	71.87
PCRM-Cosine (Ours)	**75.91**	**62.61**	**25.97**	-5.34	**72.52**

Table 3: The performance of alignment with PCRM on downstream tasks.

β_1	β_3	Accuracy with $\mathrm{Sim}_{\mathrm{cos\_sim}}(\cdot)$	Accuracy with $\mathrm{Sim}_{\mathrm{len\_rat}}(\cdot)$
10	-3	54.79	55.87
10	-5	55.73	**55.87**
10	-7	53.92	55.33
10	-9	52.26	51.60
20	-13	53.73	55.33
20	-15	**56.53**	55.73
20	-17	54.80	55.47
20	-19	55.20	55.20
30	-25	54.80	55.60
40	-35	55.60	55.07

Table 4: The performance of PCRM with different constraints measured by accuracy. The range and the numerical size of the constraints are controlled with different pairs of β_1, β_3, and β_2 is fixed to 0.001 for computational stability.

The importance of meaningful constraints cannot be overstated. Our tests with PCRM using both random and fixed constraints revealed that these constraints were either marginally effective or even detrimental to alignment progress. Furthermore, we found that prior constraints enriched with additional information yielded better results. This is evident from comparing the effectiveness of length ratio and cosine similarity in calculating similarity; the superior performance of cosine similarity can be attributed to its ability to capture implicit prior semantic information of sentences, whereas the length ratio approach is more superficial and offers less valuable information.

5.6 Results of Summarization

Except for the dialogue task, we achieve similar results with our PCRM method on the summarization task as shown in Table 3. Specifically, PCRM with cosine similarity outperforms RLHF by +0.87 points on ROUGE-L and +1.99 points on GPT-4 win rate. As to PCRM with length ratio, it outperforms RLHF by +0.27 points on BART Score and +1.34 points on GPT-4 win rate. It is obvious that there is a misalignment between ROUGE-L and BART Score- the models with high BART Score may not necessarily achieve high ROUGE-L scores. We attribute this phenomenon to the probable low correlation with human judgments, which is also reported in Wang et al. (2023a).

5.7 Analysis

5.7.1 Performance of the Variety Range and Numerical Size of Constraints on PCRM

The variety range and numerical size of constraints are two key factors that control the maximum score margin and the strength of constraints. Therefore, we conduct experiments to study the impact of the

Method	PandaLM	GPT-4 Win
SFT	65.75	55.77
DPO	75.64	60.96
PCDPO-Length (Ours)	77.02	61.11
PCDPO-Cosine (Ours)	**78.11**	**63.91**

Table 5: The experiment results of PCDPO.

variety range and the numerical size of constraints. Specifically, we explore with β_2 fixed to 0.001 in Equation 19 for computational stability, and different β_1, β_3, for variety range and numerical size on PCRM. The results are summarized in Table 4. From the results, we can observe that the unsuitable constraints may hurt the performance. We conjecture that the larger one may weaken the effect of distribution control, and the smaller one may conflict with the typical optimization. Based on these experimental results, we find that the optimal β_1 and β_3 are 20 and -15 for the cosine similarity constraint; the optimal β_1 and β_3 are 10 and -5 for the length ratio constraint. Note that β_2 prevents the occurrence of a similarity $\text{Sim}(\cdot)$ of 0, so we do not tune it and simply set a relatively small value.

5.7.2 Integrating PCRM to DPO

Rafailov et al. (2024) proposes a direct preference optimization (DPO) method that bypasses the reward modeling step and directly optimizes an LLM using preference data. However, this method still performs optimization based on ranking loss, which has the same limitation during training with preference data. Therefore, we attempt to integrate the proposed PCRM into DPO, constraining the direct preference optimization. Suppose we have the LLM trained by SFT π_θ^{SFT}, and based on it, we fine-tune the model with DPO and preference data $(y_1, y_2, x) \sim \mathcal{D}$ where $y_1 \succ y_2$. Rafailov et al. (2024) points out that the reward model can be represented as follows with some mathematical transformation on the optimization objective of PPO:

$$\pi_\theta^{\text{RM}}(y, x) = \beta \log \frac{\pi_\theta^{\text{RM}}(y|x)}{\pi_\theta^{\text{SFT}}(y|x)} + \beta \log Z(x) \tag{24}$$

where $Z(x) = \Sigma_y \pi_\theta^{\text{SFT}}(y|x) \exp\left(\frac{1}{\beta}\pi_\theta^{\text{RM}}(y|x)\right)$. Bring the above equation into Equation 16 and 18, we can obtain the loss of **P**rior **C**onstraints-based **DPO** (**PCDPO**):

$$\begin{aligned}
\mathcal{L}_{\text{PCDPO}} = &- \log P_{\pi_\theta^{\text{RM}}}\left(\Delta^* > \Delta_{\pi_\theta^{\text{RM}}} > 0\right) && (25)\\
= &- \log \sigma\left(\Delta^* - \Delta_{\pi_\theta^{\text{RM}}}\right) - \log \sigma\left(\Delta_{\pi_\theta^{\text{RM}}}\right) && (26)\\
= &- \log \sigma\left(\Delta^* - \beta \log \frac{\pi_\theta^{\text{RM}}(y_1|x)}{\pi_\theta^{\text{SFT}}(y_1|x)} + \beta \log \frac{\pi_\theta^{\text{RM}}(y_2|x)}{\pi_\theta^{\text{SFT}}(y_2|x)}\right) \\
&- \log \sigma\left(\beta \log \frac{\pi_\theta^{\text{RM}}(y_1|x)}{\pi_\theta^{\text{SFT}}(y_1|x)} - \beta \log \frac{\pi_\theta^{\text{RM}}(y_2|x)}{\pi_\theta^{\text{SFT}}(y_2|x)}\right) && (27)
\end{aligned}$$

With the theory above, we conduct experiments with the constraints on the alpaca dataset. The experimental results are shown in Table 5. From the results, we can find that PCRM can yield consistency improvements in DPO. Notably, when armed with the cosine similarity constraint, we can obtain a +2.95 points improvement on the GPT-4 win rate.

5.7.3 Inference with Different Temperature

It is known to us that temperature influences the performance of inference, so we make inferences with different sampling temperatures on the dialogue task three times and calculate their mean in case of random variations while comparing different methods. Results are reported in Figure 2.

The experiment results are consistent with the previous. Our method, PCRM with cosine similarity, outperforms the vanilla reward model no matter which sampling temperature is used, and the increase

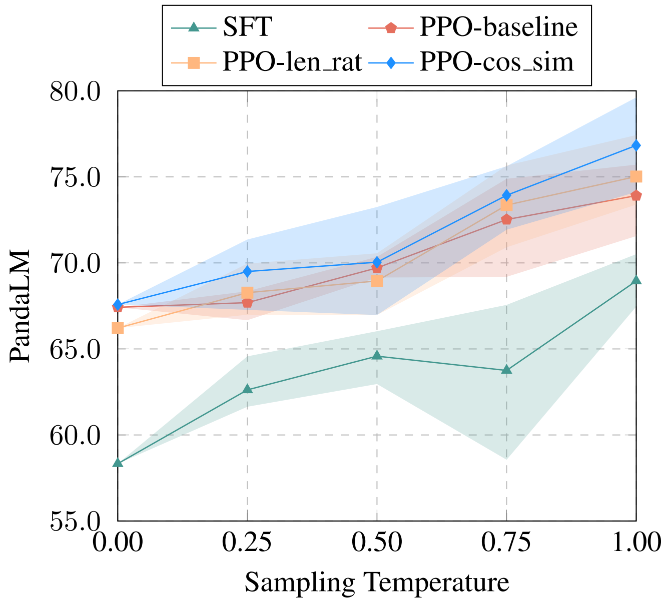


Figure 2: PandaLM scores for different sampling temperatures using different methods. For each dialogue model, we conduct the generation three times and report the mean score of these generated responses.

of PCRM with length ratio outperforms most of the temperatures. As analyzed before, we infer that the performance of the length ratio is limited due to the easy-to-learn prior information.

6 Conclusions

RLHF plays an important part in applications based on LLMs for their ability to align language models with human preferences. As a proxy for human preferences, the reward model makes a significant contribution. We have introduced PCRM, a method of training reward models with prior constraints to better control the distribution of reward scores without sacrificing prediction accuracy. Instead of optimizing the vanilla reward model with the goal of maximizing the margin of scores, PCRM restricts the optimization process with the provided prior information and fits the distribution of reward scores for better alignment. Furthermore, this method is applicable to DPO, which treats the language model as a reward model and optimizes it directly. Through experiments on downstream tasks, we have validated the effectiveness of this method.

7 Limitations

Our results raise several questions that are beyond the scope of the present study: How can we determine the appropriate range of constraints for different tasks or datasets? For example, we explore hyperparameters in dialogue and summarization tasks, what about when it comes to a brand new task? Can the function of the constraints be replaced with any other prior information, and if so, will that be effective? Finally, it would be advantageous if the prior constraints could be learned automatically from data, rather than being manually set.

Acknowledgements

This work was supported in part by the National Science Foundation of China (No.62276056), the Natural Science Foundation of Liaoning Province of China (2022-KF-16-01), the Fundamental Research Funds for the Central Universities (Nos. N2216016 and N2316002), the Yunnan Fundamental Research Projects (No. 202401BC070021), and the Program of Introducing Talents of Discipline to Universities, Plan 111 (No.B16009).

References

Afra Amini, Tim Vieira, and Ryan Cotterell. 2024. Direct preference optimization with an offset. *arXiv preprint arXiv:2402.10571*.

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.

Pengyu Cheng, Jiawen Xie, Ke Bai, Yong Dai, and Nan Du. 2023. Everyone deserves a reward: Learning customized human preferences. *arXiv preprint arXiv:2309.03126*.

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

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2024. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36.

Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, et al. 2023. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

R Duncan Luce. 2005. *Individual choice behavior: A theoretical analysis*. Courier Corporation.

Do June Min, Veronica Perez-Rosas, Kenneth Resnicow, and Rada Mihalcea. 2024. Dynamic reward adjustment in multi-reward reinforcement learning for counselor reflection generation. *arXiv preprint arXiv:2403.13578*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

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

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

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Chenglong Wang, Hang Zhou, Kaiyan Chang, Tongran Liu, Chunliang Zhang, Quan Du, Tong Xiao, and Jingbo Zhu. 2023a. Learning evaluation models from large language models for sequence generation. *arXiv preprint arXiv:2308.04386*.

Chenglong Wang, Hang Zhou, Yimin Hu, Yifu Huo, Bei Li, Tongran Liu, Tong Xiao, and Jingbo Zhu. 2023b. Esrl: Efficient sampling-based reinforcement learning for sequence generation. *arXiv preprint arXiv:2308.02223*.

Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, et al. 2023c. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. *arXiv preprint arXiv:2306.05087*.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2024. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems*, 36.

Tong Xiao and Jingbo Zhu. 2023. Introduction to transformers: an nlp perspective. *arXiv preprint arXiv:2311.17633*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*.

Yifan Zhong, Chengdong Ma, Xiaoyuan Zhang, Ziran Yang, Qingfu Zhang, Siyuan Qi, and Yaodong Yang. 2024. Panacea: Pareto alignment via preference adaptation for llms. *arXiv preprint arXiv:2402.02030*.

Banghua Zhu, Jiantao Jiao, and Michael I Jordan. 2023. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. *arXiv preprint arXiv:2301.11270*.

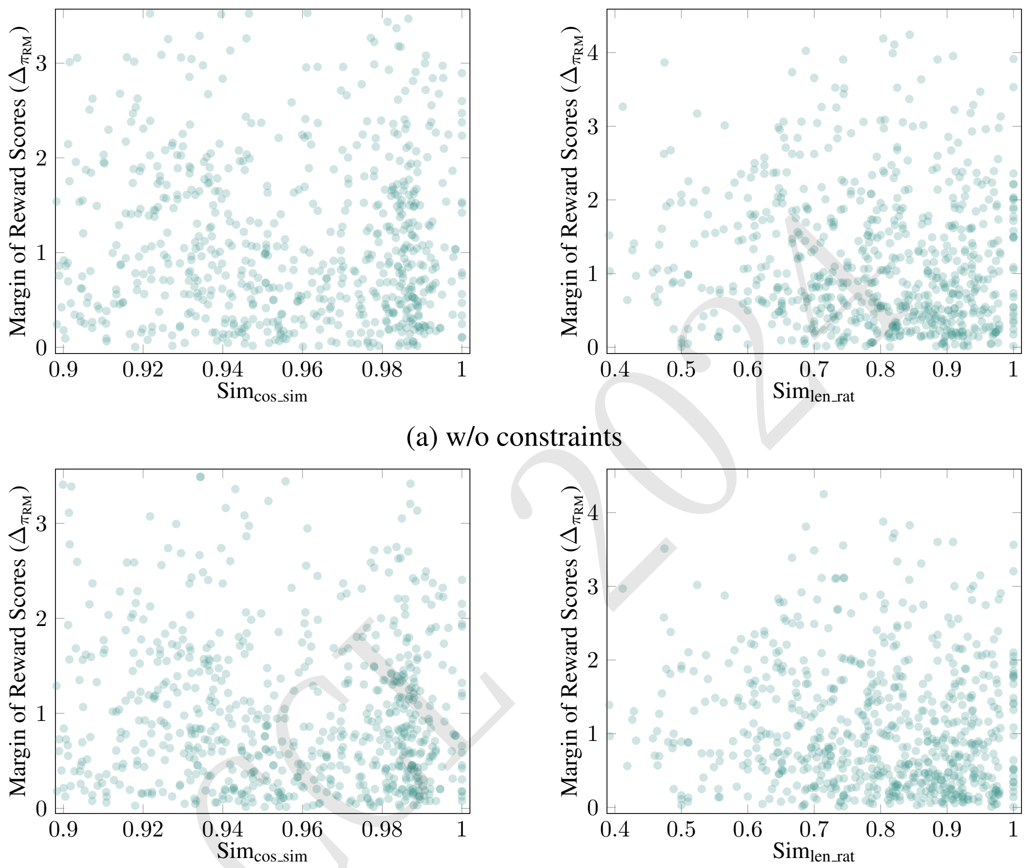
Appendix A. Calculating Accuracy of the Reward Scores

Suppose we have a test set for human preference $(x, y_1, y_2) \sim \mathcal{D}_{\text{test}}$, in which y_1 is preferred than y_2 with the same x by human, and the corresponding scores predicted by the reward model are $\pi_\theta^{\text{RM}}(y1, x)$, $\pi_\theta^{\text{RM}}(y2, x)$. The accuracy of the scores is defined as:

$$\text{Acc}(\pi_\theta^{\text{RM}}, \mathcal{D}_{\text{test}}) = \frac{\text{Count}_{(x,y_1,y_2)\sim\mathcal{D}_{\text{test}}}\left(\pi_\theta^{\text{RM}}(y1, x) > \pi_\theta^{\text{RM}}(y2, x)\right)}{\text{Count}\left((x, y_1, y_2) \in \mathcal{D}_{\text{test}}\right)} \tag{28}$$

where $\text{Count}(\cdot)$ denotes the total number of the samples that meet the condition.

Appendix B. Distribution of Reward Scores for Summarization Task



(a) w/o constraints

(b) w/ constraints

Figure 3: The distribution between the margin of predicted reward scores and the similarity of paired data (calculated by cosine similarity or length ratio) with or without constraint on the summarization task.