

Preference Ranking Optimization for Human Alignment

Feifan Song¹, Bowen Yu^{2*}, Minghao Li²

Haiyang Yu², Fei Huang², Yongbin Li², Houfeng Wang^{1*}

¹National Key Laboratory of Multimedia Information Processing, Peking University

²Alibaba Group

songff@stu.pku.edu.cn

{yubowen.ybw, lmh397008, yifei.yhy, shuide.lyb}@alibaba-inc.com

wanghf@pku.edu.cn

Abstract

Large language models (LLMs) often contain misleading content, emphasizing the need to align them with human values to ensure secure AI systems. Reinforcement learning from human feedback (RLHF) has been employed to achieve this alignment by combining a reward model, typically based on Bradley-Terry paired comparison, with an RL algorithm such as Proximal Policy Optimization (PPO) to optimize LLM responses. However, RLHF exhibits complexity, instability, and sensitivity to hyperparameters. In this paper, we propose Preference Ranking Optimization (PRO) as an alternative to PPO for directly aligning LLMs with the Bradley-Terry comparison. PRO extends the pairwise Bradley-Terry comparison to accommodate preference rankings of any length. By iteratively contrasting the likelihood of generating responses, PRO instructs the LLM to prioritize the best response while progressively ranking the remaining responses. In this manner, PRO effectively transforms human alignment into aligning the probability ranking of n responses generated by LLM with the preference ranking of humans towards these responses. Experiments have shown that PRO outperforms existing alignment algorithms, achieving comparable results to ChatGPT and human responses through automatic-based, reward-based, GPT-4, and human evaluations. Furthermore, we demonstrate that longer, more diverse, and higher-quality preference ranking sequences can consistently enhance the performance of human alignment¹.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in meeting the diverse information needs of users (Brown et al., 2020b; Chowdhery et al., 2022; Bubeck et al., 2023; Touvron

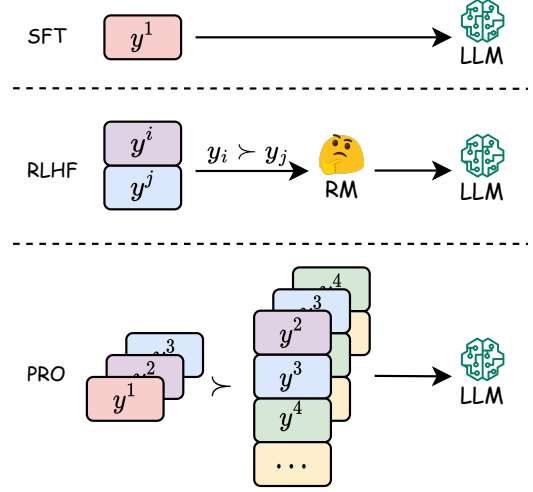


Figure 1: Comparison among different human alignment paradigms. SFT utilizes just the most preferred response y^1 while RLHF first samples candidates $y^i > y^j$ from the whole ranking to train a reward model, then relies on it to fine-tune the agent LM. The proposed PRO instead distinguishes y^i against all members from the sub-ranking y^1, \dots, y^n .

et al., 2023; Li et al., 2023), primarily attributed to the extensive range of information sources integrated into their pretraining datasets (Laurençon et al., 2022; Muennighoff et al., 2023). Nevertheless, despite leveraging the extensive global knowledge and human behavior encoded within their trillion-token pretraining corpus, LLMs are unavoidably impacted by the existence of misleading, toxic, and detrimental content encompassed within it (Bai et al., 2022b; Ouyang et al., 2022b). Consequently, *aligning LLMs to human values, by selecting human-preferred responses from the vast response space of LLMs* (Rafailov et al., 2023), becomes pivotal in constructing AI systems that are secure, efficient, and manageable for deployment across numerous applications (Peng et al., 2023).

Several studies have employed reinforcement learning from human feedback (RLHF) to achieve

* Corresponding author.

¹The code of this work is available at <https://github.com/AlibabaResearch/DAMO-ConvAI/tree/main/PRO>

this goal (Stiennon et al., 2020a; Xue et al., 2023). RLHF involves fitting a reward model to human preferences, employing the Bradley-Terry paired comparison (Bradley and Terry, 1952). Bradley-Terry seeks to assign higher scores to preferable responses in comparison to unfavorable ones when presented with the same prompt. The RL algorithm, specifically PPO (Schulman et al., 2017), is then utilized to optimize an LLM for generating high-reward responses (Akyürek et al., 2023). This approach offers two notable advantages over supervised fine-tuning. Firstly, it has the capability to utilize both positively and negatively labeled responses (Zhang et al., 2023). Secondly, it can engage in self-bootstrapping to rectify the model’s inadequate responses (Kwon et al., 2023). Although impressive, the RLHF pipeline is significantly more complex than supervised learning, prone to optimization instability, and sensitive to hyperparameters (Rafailov et al., 2023; Wu et al., 2023; Yuan et al., 2023). These limitations arise mainly from employing the PPO algorithm to align the LLM with the reward model’s preferences. However, the reward model itself aims to optimize the Bradley-Terry paired comparison. This prompts an important research question: Is it possible to bypass the requirement for PPO and enable direct learning of the Bradley-Terry comparison by the LLM?

In this paper, we propose **Preference Ranking Optimization (PRO)** as a replacement for PPO, providing an exceptionally exciting answer to this question. We first extend the pairwise comparison of Bradley-Terry to encompass comparisons within preference rankings of arbitrary lengths. Let us assume that given a prompt x , we have access to a set of ranked responses represented as y^1, y^2, \dots, y^n . The PRO algorithm begins by teaching the LLM to treat the best response y^1 as the positive and treat the remaining responses as negatives by contrasting generation likelihood. This prioritization implies that the likelihood of generating a reply by LLM is significantly higher compared to generating other responses that humans consider inferior. It then iteratively removes the current response and proceeds to the next one. This process is repeated until there are no responses that perform worse than the current response, which indicates reaching y^n and sufficiently imposing the desired ranking preferences. PRO aims to achieve a probability ranking of n responses generated by the LLM that aligns with human preference ranking. As n ap-

proaches infinity, we can consider the output space of LLM to be perfectly aligned with human preferences. Specifically, when $n = 2$, PRO effectively optimizes the LLM using the Bradley-Terry Comparison method.

This formulation possesses the following advantages (1) PRO allows for the complete utilization of ranking sequences of any length, unlike standard fine-tuning that only considers the best response (Zhang et al., 2023), or RLHF that relies solely on pairwise comparisons for training the reward model (Stiennon et al., 2020a). With longer ranking sequences, PRO can better approximate the goal of Human Alignment: selecting human-preferred responses from the response space of LLMs, by identifying more responses that are known to be worse than a given response in human values. (2) PRO naturally inherits the self-bootstrapping benefit of RLHF. During training, responses sampled from the LLM can be added to the response set and reranked based on their reward scores, using an additional reward model similar to RLHF. The LLM is then continuously optimized by PRO on the extended preference sequences. (3) PRO only requires the inclusion of a differentiable contrastive loss on top of standard fine-tuning, avoiding the drawbacks associated with RL’s non-differentiable optimization.

We conduct experiments on HH-RLHF, to thoroughly compare our PRO with LLaMA, Alpaca, ChatGPT, and other competitive human alignment algorithms such as BoN, CoH, RLHF, and RRHF, using various evaluation methods including automatic scoring, reward modeling, GPT-4 evaluation, and human evaluation. Our observations are as follows: (1) With a ranking length of 2, our PRO has surpassed the current competitive baselines. It outperforms SFT by 6.52 points and RRHF by 3.1 points, establishing itself as the state-of-the-art alignment algorithm. (2) The longer the ranking length in human preference ranking, the better human alignment, and the more prominent the performance improvement of PRO. For instance, by adding responses generated by ChatGPT to the dataset and increasing the ranking length to 3, PRO achieves a 4.14-point improvement over BoN and a 4.85-point improvement over RRHF, with a reward score similar to ChatGPT, but with only 7B parameters. (3) The higher the quality and diversity of the candidates in the preference ranking sequence, the better the performance of PRO. (4) The

performance gain from self-bootstrapping is lower compared to adding high-quality outputs generated by other LLMs to the preference ranking sequence.

2 Preliminary

We commence by providing a brief review of RLHF. In order to train LLM to generate responses that align with human preferences, RLHF consists of three stages, which are outlined as follows:

The first stage is Supervised Fine-tuning (SFT): Labelers furnish the desired behavior’s response with t tokens, denoted as $y = y_1, \dots, y_t$, for a given input prompt, denoted as x . Subsequently, RLHF proceeds to fine-tune a pre-trained LLM using supervised learning (maximum likelihood) on this data, resulting in a model denoted as π_{SFT} :

$$\mathcal{L}_{\text{SFT}} = - \sum_t \log P_{\pi_{\text{SFT}}}(y_t | x, y_1, \dots, y_{t-1}). \quad (1)$$

In the second phase, the SFT model is utilized by providing prompts x to generate pairs of responses. These pairs are then presented to human labelers, who express their preferences by indicating a favored answer as y^1 , while the other response is denoted as y^2 . Specifically, we have $y^1 \succ y^2 \mid x$ to represent the preferences of human labelers. To predict these preferences, the previous work employ the Bradley-Terry (BT) model, which defines the preference probability as follows:

$$P_{\text{BT}} = \frac{\exp(r_\phi(x, y^1))}{\exp(r_\phi(x, y^1)) + \exp(r_\phi(x, y^2))} \quad (2)$$

This objective is framed as a binary classification problem to train the reward model: $\mathcal{L}_{\text{BT}} = -\log \sigma(r_\phi(x, y^1) - r_\phi(x, y^2))$, where σ is the logistic function, r_ϕ is the reward model. During the third phase, RLHF utilizes the acquired r_ϕ to provide feedback to π_{SFT} . Specifically, RLHF formulates the following optimization problem:

$$\max_{\pi_\theta} \mathbb{E} \left(r_\phi(x, y) - \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{SFT}}(y \mid x)} \right) \quad (3)$$

Here, β controls the deviation from the base reference policy π_{SFT} to maintain generation diversity and prevent from generating of only high-reward but meaningless answers. It is important to note that this objective is non-differentiable and is typically optimized using reinforcement learning methods, such as Proximal Policy Optimization (PPO) (Schulman et al., 2017). Consequently,

RLHF has been criticized for several drawbacks, including increased complexity compared to supervised learning, sensitivity to hyperparameters, and the requirement for additional training of reward models and value networks.

3 Methodology

In this section, we first derive the evolution from the Bradley-Terry comparison to our proposed PRO, achieving a shift from reward model-oriented preference alignment to aligning the probabilistic ranking of n responses generated by the LLM with the human preference ranking. This alignment helps to avoid the numerous drawbacks associated with RLHF. Furthermore, we demonstrate the flexibility of our PRO in its ability to integrate with the reward model, thereby attaining advantages such as affordable preference ranking, differentiated contrast, and self-bootstrapping augmentation.

3.1 From RLHF to PRO

Upon re-evaluating the process of RLHF mentioned above, it becomes evident that the criticized shortcomings of RLHF stem from its utilization of a reward model as a learning proxy. Therefore, if we eliminate the proxy and directly optimize the LLM to learn the objective of the reward model, can we circumvent the aforementioned challenges?

For this purpose, we re-examine the objective of the reward model, Bradley-Terry (Equation 2), which aims to make the model understand $y^1 \succ y^2$ through score comparison. For a given prompt x , assuming that there are only two responses, y^1 and y^2 , in the LLM response space, the reward model should prefer y^1 . Naturally, if we expand the response space of the LLM, for example, there exist n possible responses $\{y^i\}$, and the human-annotated order $y^{1,\dots,n}$ is $y^1 \succ y^2 \succ \dots \succ y^n$. We define the partial order between y^1 and candidates behind it as $y^{1,2:n} = y^1 \succ \{y^2, \dots, y^n\}$, then the objective of Bradley-Terry becomes:

$$P(y^{1,2:n} \mid x) = \frac{\exp(r(x, y^1))}{\sum_{i=1}^n \exp(r(x, y^i))} \quad (4)$$

Furthermore, it is important to acknowledge that this objective does not fully leverage the rankings $y^{1,\dots,n}$ since it only characterizes $y^1 \succ y^2, \dots, y^n$, disregarding the $n - 2$ valuable rankings such as $y^2 \succ y^3, \dots, y^n$ and $y^{n-1} \succ y^n$. Consequently, we

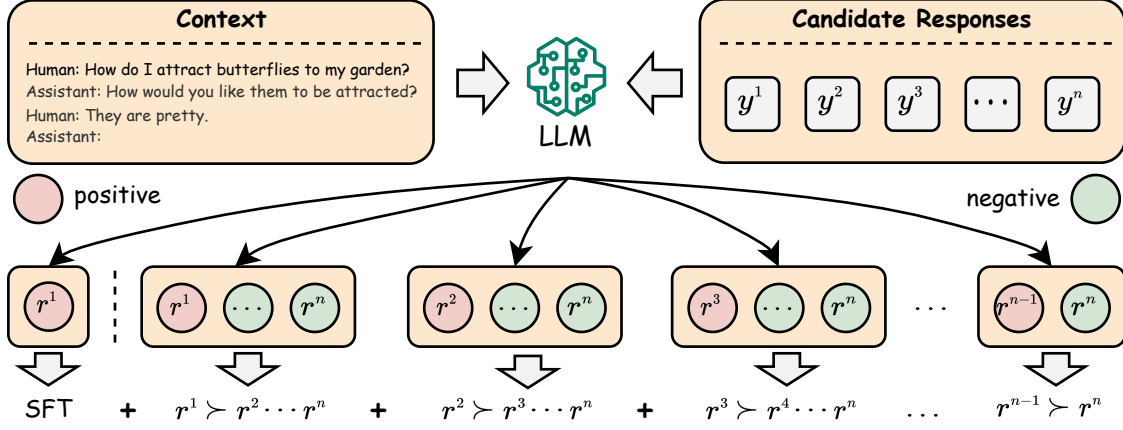


Figure 2: The pipeline of PRO for Human Feedback Alignment learning. Each candidate is concatenated with the prompt first, then processed by the LLM to estimate corresponding rewards, which are optimized by Equation 7.

propose an extension to Equation 4 as follows:

$$\begin{aligned}
 P(y^{1,\dots,n} | x) &= \prod_{k=1}^{n-1} P(y^{k,k+1:n} | x) \\
 &= \prod_{k=1}^{n-1} \frac{\exp(r(x, y^k))}{\sum_{i=k}^n \exp(r(x, y^i))} \quad (5)
 \end{aligned}$$

To impose the desired ranking presented by $y^1 \succ y^2 \succ \dots \succ y^n$, we use Equation 4 in a recursive manner where we start with the first response, treat the remaining responses as negatives, drop the current response, and move to the next. We repeat this procedure until there are no responses left. Surprisingly, this objective aligns closely with the ultimate goal of Human alignment, which is the task of selecting desired responses from the vast response space of LLMs (Rafailov et al., 2023). In other words, if $n \rightarrow \infty$, then Equation 5 is able to exhaustively explore all possible responses and annotate y^1 as the most desired response, thus perfect alignment with humans.

Based on this motivation, we propose the Preference Ranking Optimization (PRO) algorithm. Instead of optimizing the LLM to approximate the Reward Model, we propose directly training the LLM to reach Equation 5. Figure 2 demonstrates the pipeline of PRO algorithm. Specifically, we define $r_{\pi_{\text{PRO}}}(x, y^k)$ as the function parameterized by our desired LLM π_{PRO} :

$$r_{\pi_{\text{PRO}}}(x, y^k) = \frac{1}{|y^k|} \sum_{t=1}^{|y^k|} \log P_{\pi_{\text{PRO}}}(y_t^k | x, y_{<t}^k) \quad (6)$$

The LLM π_{PRO} computes the score of response y^k by multiplying the probabilities of each token

generated by π_{PRO} itself. When Equation 5 is fully optimized, π_{PRO} is able to consistently generate the most preferred response (with a higher output probability) from a candidate set, thereby capturing human preferences. In addition to adhering to human preferences, it is also desirable for the model to generate fluent replies. Therefore, we incorporate the original supervised loss (Equation 1) that requires the model to fit the responses considered the best by humans. Consequently, the overall optimization objective can be summarized as follows:

$$\mathcal{L}(y^{1,\dots,n} | x) = \mathcal{L}_{\text{PRO}} + \beta \mathcal{L}_{\text{SFT}} \quad (7)$$

where \mathcal{L}_{SFT} is the NLL loss of the top 1 candidate and β is the hyper-parameter to maintain the balance between text quality and human preference. \mathcal{L}_{PRO} is defined as:

$$\mathcal{L}_{\text{PRO}} = - \sum_{k=1}^{n-1} \log \frac{\exp(r_{\pi_{\text{PRO}}}(x, y^k))}{\sum_{i=k}^n \exp(r_{\pi_{\text{PRO}}}(x, y^i))} \quad (8)$$

By comparing Equation 7 and Equation 3, it can be observed that, at the training objective level, PRO and RLHF have similar structures, but PRO is more efficient. Both PRO and RLHF share the primary goal of human alignment. RLHF achieves this by providing better responses through a reward model with higher discrete scores, requiring RL techniques. In contrast, PRO directly achieves this through ranking scores, thereby avoiding many drawbacks associated with RL. The second objective of both PRO and RLHF is to ensure high-quality model outputs. PRO’s alignment objective is differentiable, allowing for multi-task learning by combining alignment and SFT objectives through

single-stage training. On the other hand, RL, due to the discrete optimization problem, requires training the SFT model first and then constraining the RL model from deviating excessively from SFT. Consequently, RLHF necessitates two-stage training, which undoubtedly increases training costs.

Comparing Equation 1, Equation 7, and Equation 2, we can observe that PRO is more data-efficient. SFT can only leverage responses considered as desired in a preference ranking, completely disregarding negative responses. We believe negative examples are crucial in human alignment since LLM should not only learn what is good but also discern what is not. The critical component of RLHF, the reward model, is trained through pairwise response comparisons, requiring $\binom{2}{n}$ comparisons for a ranking of length n . In contrast, PRO only needs $n - 1$ comparisons and introduces more negative examples in each comparison compared to RLHF. Therefore, PRO provides better and more stable score estimates since more negative examples enlarge the response space, making the ranking process for obtaining the desired response more aligned with human expectations.

We also observe that PRO establishes a bridge between human alignment and contrastive learning. Contrastive learning has recently shown significant advancements in the field of self-supervised learning (Oord et al., 2018), where the main objective is to maximize the similarity between a query and its corresponding positive instance, while creating a distance from other negatives (Sohn, 2016). In the context of PRO, we model similarity as the parameters of the language model to measure the likelihood of generating a response. We expect that this modeling approach will encourage future researchers to fully explore the extensive research achievements in contrastive learning, ultimately achieving better human alignment.

3.2 Grafting RLHF onto PRO

While PRO can optimize directly on the human-annotated preference ranking sequence without the need for introducing concepts like the reward model in RLHF, we have found that grafting RLHF onto PRO can bring more flexibility to PRO. We outline three possible upgrades as follows:

Affordable Preference Ranking. PRO is highly flexible, relying solely on an arbitrarily long ranked preference sequence. The source of the sequence is unrestricted, allowing for various possibilities.

One approach involves requesting annotators to imagine multiple responses of different quality. Alternatively, a more efficient method entails utilizing different existing LLMs, such as ChatGPT and Alpaca, to generate multiple responses. These responses can then be ranked using an additional reward model r_ϕ , similar to RLHF.

Differentiated Contrast. The formulation of \mathcal{L}_{PRO} , as shown in Equation 8, treats all responses $y^i \prec y^k$ as negative examples of y^k and applies the same penalty to them. However, this approach may not be reasonable, especially when the preference scores of different y^i are similar. For instance, when the preference of y^{k+1} is only slightly worse than y^k , while y^n is significantly worse than y^k , the model should differentiate and apply different penalty strengths, slightly penalizing y^{k+1} and heavily penalizing y^n compared to y^k . To address this, we propose using the score $r_\phi(x, y^i)$ from a reward model r_ϕ to indicate the numerical preference of y^i , and modify Equation 8 as follows:

$$\mathcal{L}_{\text{PRO}} = - \sum_{k=1}^{n-1} \log \frac{\exp \left(\frac{r_{\pi_{\text{PRO}}}(x, y^k)}{\mathcal{T}_k^k} \right)}{\sum_{i=k}^n \exp \left(\frac{r_{\pi_{\text{PRO}}}(x, y^i)}{\mathcal{T}_k^i} \right)} \quad (9)$$

where

$$\mathcal{T}_k^{i>k} = \frac{1}{r_\phi(x, y^k) - r_\phi(x, y^i)} \quad (10)$$

$$\mathcal{T}_k^k = \min_{i>k} \mathcal{T}_k^i \quad (11)$$

When the difference between $r_\phi(x, y^k)$ and $r_\phi(x, y^i)$ increases, the preference gap between y^k and y^i becomes more evident. Consequently, the temperature \mathcal{T}_k^i decreases, amplifying the penalty of positive example y^k towards y^i , while it decreases when the difference is smaller. \mathcal{T}_k^k is defined as the minimum temperature among all the negative examples to maintain a balance between the numerator and denominator. Our experiments (§4.7) reveal that the dynamic temperature design significantly enhances model performance when optimizing \mathcal{L}_{PRO} alone while excluding \mathcal{L}_{SFT} . It also provides some performance gains when jointly optimizing \mathcal{L}_{PRO} and \mathcal{L}_{SFT} .

Self-bootstrapping Augmentation. Furthermore, it is worth noting that the length of sequences that PRO relies on is variable. In other words, there is no requirement for fixed sequences during training. This allows us to consider grafting the self-bootstrapping advantage of RLHF as

a subset onto PRO. Specifically, RLHF aims to continuously evaluate the model’s responses during training by employing a reward model. Positive or negative rewards are provided to bootstrap the Language Model itself. Similarly, with PRO, given the prompt x and the current model, we sample a response \hat{y} and add it to the existing response set $\{y^1, \dots, y^n\}$. Subsequently, we re-rank the responses using the reward model, yielding $p(\hat{y}^{1, \dots, n+1} | x)$. Therefore, further optimization can be performed by refreshing Equation 7:

$$\mathcal{L}_{\text{PRO}}(y^{1, \dots, n} | x) \Rightarrow \mathcal{L}_{\text{PRO}}(\hat{y}^{1, \dots, n+1} | x) \quad (12)$$

The abstract training procedures are as follows:

Algorithm 1: Self-bootstrap PRO

Input: Language Model π_{LM}^0 , Reward Model r_ϕ , Raw Dataset D ,
Output: The fine-tuned LM π_{PRO}

- 1 Split D into $\{D_0, D_1, \dots, D_{K-1}\}$
- 2 **for** $D_i \in \{D_0, D_1, \dots, D_{K-1}\}$ **do**
- 3 **for** Sample $d \in D_i$ **do**
- 4 $x \leftarrow \text{Prefix}(d)$
- 5 $\{y^j\} \leftarrow \text{Candidates}(d)$
- 6 $\hat{y} \leftarrow \pi_{\text{LM}}^i(x)$ // Sampling from LM
- 7 Add \hat{y} to $\{y^j\}$
- 8 Score and re-rank $\{y^j\}$ with x and r_ϕ
- 9 **end for**
- 10 $\pi_{\text{LM}}^{i+1} \leftarrow \text{PRO}(\pi_{\text{LM}}^i, D_i)$ // Train LLM
- 11 **end for**
- 12 $\pi_{\text{PRO}} \leftarrow \pi_{\text{LM}}^K$

4 Experiments

4.1 Datasets

We conduct experiments mainly based on **Human Preference Data about Helpfulness and Harmlessness**, i.e., HH-RLHF described in Bai et al. (2022a). It has 4 sub-sets, namely Harmless_{base}, Helpful_{base}, Helpful_{online} and Helpful_{rejection}, where each sample contains two different conversations rated by human annotators and is grouped into train/test splits. We refer to the code² released by OpenAssistant and filter all data to ensure that the chosen and rejected conversations in the same sample have identical contexts but different responses. Details can be found in Table 1.

We combine training data from 4 sub-sets to fine-tune models and evaluate them on each of the test sets, while we do validation with 280 samples randomly selected from all test data. Each sample

from the raw dataset contains a chosen conversation and a rejected one, which constitutes a relatively short ranking. To further evaluate the performance of different models on longer human preference rankings, we enhance each sample with additional responses from Alpaca (Taori et al., 2023) and ChatGPT³, thereby expanding the range of ranked candidates. We refer to these augmented datasets as $HH\text{-}RLHF_{\text{LLM},i}$, where LLM represents the language models used (Alpaca, ChatGPT, etc.), and i denotes the length of the rankings. The unmodified dataset is referred to as $HH\text{-}RLHF_{\text{raw}}$.

4.2 Evaluation Metrics

We present the findings of our study using various evaluation methods: automatic, model-based, and human-based metrics. In our main experiment, we utilize BLEU (Papineni et al., 2002) to assess the text quality and the Reward model to measure the level of human preference gained. These metrics allow us to evaluate the performance of numerous models automatically. For the analysis experiment, we employ human evaluators to conduct pairwise comparisons among the top-performing models identified through automated evaluations. Human evaluation is the gold standard for assessing human preferences (Zhou et al., 2023). An annotator judge is presented with a question and two responses and tasked with determining the better option or declaring a tie. Furthermore, recent studies have shown that GPT-4 (OpenAI, 2023) effectively evaluates the responses of chat assistants and aligns with human preferences (Zheng et al., 2023; Wang et al., 2023). Consequently, we involve GPT-4 to select a model generation from the two options. To mitigate positional bias (Zheng et al., 2023; Wang et al., 2023), we evaluate each candidate in both positions during two separate runs, and the final score is computed as the average of the two runs.

4.3 Implementation Detail

In this work, we choose LLaMA-7B (Touvron et al., 2023) as the backbone model, which has become a widespread test field for LLM research (Taori et al., 2023). We fine-tune it with PRO algorithm built on Huggingface.Library (Wolf et al., 2020).

We calculate BLEU scores to compare inference results with human-selected responses in test sets. To capture human preferences, reward models are

²<https://github.com/LAION-AI/Open-Assistant>

³<https://chat.openai.com/>

Sub-set		# train	# test
Harmless _{base}	Raw	42537	2312
	Filtered	42536	2312
Helpful _{base}	Raw	43835	2354
	Filtered	43835	2354
Helpful _{online}	Raw	22007	1137
	Filtered	22002	1137
Helpful _{rejection}	Raw	52421	2749
	Filtered	52420	2749

Table 1: Statistics of $HH\text{-}RLHF_{\text{raw}}$.

used as proxies. Additionally, we expand the training set by incorporating output results from existing LLMs, requiring the sorting of the expanded preference rankings. However, manual sorting is time-consuming and costly, especially considering the large number of instances in the training set. Therefore, we employ an additional reward model to score and rearrange all candidate rankings during the pre-processing stage of training. To avoid unfairness in evaluation, we select two different reward models for training and evaluation, which we denote as RM_{train} ⁴ and RM_{eval} ⁵, respectively. Reward values from RM_{eval} are normalized with the Sigmoid function in case RM_{eval} provides extreme values that excessively influence the overall performance.

Moreover, we assign β , the weight SFT loss, to $0.05 * (l - 1)^2$ where l is the ranking length. The sequence length, epoch, and learning rate are set to 512, 2, and $5e-6$, respectively, while the maximum number of new tokens generated during inference is 128. We deploy our complete framework using 8 devices, 7 of which are dedicated to the model training, while the remaining one houses RM_{train} for validation and potential self-bootstrapping augmentation (We consider this augmentation strategy as an analytical experiment and, unless otherwise specified, augmentation will not be enabled). With a batch size of 2 per device, we leverage a gradient accumulation step of 8, resulting in a total batch size of 112. More particulars can be found in our code.

⁴<https://huggingface.co/OpenAssistant/oasst-rm-2.1-pythia-1.4b-epoch-2.5>

⁵<https://huggingface.co/OpenAssistant/oasst-rm-2-pythia-6.9b-epoch-1>

4.4 Baselines

We compare PRO with zero-shot baselines, and models fine-tuned on LLaMA-7B (Touvron et al., 2023) which share the same backbone with PRO: **LLaMA** (Touvron et al., 2023) is a collection of prevalent foundation models released to enhance research on LLM techniques of training, inference, and widespread applications. We evaluate the 7B version of LLaMA (LLaMA-7B) to be consistent with other fine-tuned baselines.

Curie (Brown et al., 2020a) is considered as the 6.7B version of GPT-3, which has a similar size to LLaMA-7B. The model name used in API calls is text-curie-001.

Alpaca (Taori et al., 2023) is an instruction-tuned version of LLaMA based on 52K instruction-following data. It is estimated to have a similar instruction-following competence with text-davinci-003 on the Self-Instruct evaluation suite (Wang et al., 2022).

ChatGLM (Du et al., 2022) is a bilingual chatbot with 6.2B parameters. Having been implemented on GLM architecture (Du et al., 2022) and trained with SFT and RLHF on a large-scale conversation dataset, it manifests great potential of being in line with human preference. We implement it with its [official code](#).

ChatGPT is an online chat platform developed by OpenAI, which possesses great human-like abilities and allows versatile uses completed in the conversation form, after RLHF fine-tuning.

SFT is the basic method that naively selects the top 1 candidate to fine-tune language models. Note that if we choose the best response in a preference ranking sequence sorted by a reward model, known as best-of-n sampling, **SFT** evolves into **BoN**.

RLHF is successively promoted by Ziegler et al. (2019) and Ouyang et al. (2022a) to align the core of language models with human preference in Reinforcement Learning settings. We implement SFT and RLHF according to [trlx](#).

CoH (Liu et al., 2023) enforces language models to differentiate the most preferred candidate from the least preferred with prompts, which actually aligns models with human preference from a semantic perspective. We implement it with Huggingface.Library (Wolf et al., 2020) according to its [original version](#).

RRHF (Yuan et al., 2023) takes candidate ranking into account, and distinguishes different candidates through pair-wise ranking losses. We implement it

Training Set	Method	Harmless _{base}		Helpful _{base}		Helpful _{online}		Helpful _{rejection}		Total	
		BLEU	Reward	BLEU	Reward	BLEU	Reward	BLEU	Reward	BLEU	Reward
Zero-shot	LLaMA	10.82	51.16	12.78	31.71	15.02	38.91	14.60	34.85	13.13	38.94
	Curie	14.23	50.71	17.33	45.51	17.11	51.36	18.99	48.68	16.99	48.71
	Alpaca	15.07	53.03	19.68	49.80	18.77	55.74	22.21	53.72	19.12	52.72
	ChatGLM	15.39	63.30	20.16	59.14	30.99	61.10	25.41	61.45	21.99	61.27
	ChatGPT	15.51	71.44	21.38	65.94	29.81	67.94	26.52	68.39	22.56	68.48
<i>HH-RLHF</i> _{raw}	SFT	15.07	55.96	20.40	41.36	29.36	54.08	25.54	47.08	21.80	48.83
	RLHF	14.54	55.05	19.86	42.16	28.04	53.40	25.11	47.73	21.19	48.93
	CoH	13.34	45.47	23.17	39.03	33.84	52.63	29.79	46.57	24.06	45.00
	RRHF	13.49	53.98	18.76	48.23	30.68	56.44	24.95	52.51	20.91	52.25
	PRO	12.05	62.96	20.83	48.51	28.75	59.02	27.17	53.28	21.54	55.35
<i>HH-RLHF</i> _{Alpaca,3}	BoN	16.75	59.24	22.81	54.04	29.89	61.00	27.76	58.04	23.7	57.66
	RLHF	16.33	56.61	23.12	54.85	30.54	60.97	27.94	58.4	23.82	57.28
	CoH	13.71	47.36	22.45	42.34	33.17	53.19	28.76	48.61	23.54	47.15
	RRHF	12.79	54.18	19.21	53.23	31.53	59.04	25.14	56.76	21.02	55.39
	PRO	14.41	62.60	22.47	54.38	25.61	60.90	26.82	58.26	22.11	58.72
<i>HH-RLHF</i> _{ChatGPT,3}	BoN	15.05	67.85	20.77	60.43	31.27	64.36	26.47	63.14	22.45	63.83
	RLHF	13.63	61.97	20.12	55.29	28.89	59.78	24.65	58.26	20.99	58.65
	CoH	13.44	56.87	21.89	51.52	34.04	59.51	28.24	56.35	23.26	55.58
	RRHF	13.02	64.63	18.95	61.38	31.37	63.26	24.75	63.28	20.86	63.12
	PRO	15.53	73.08	22.30	64.78	29.35	66.66	27.49	66.95	23.07	67.97

Table 2: Main Results. PRO consistently acquires more reward than all fine-tuned baselines, while is close to or even exceeding ChatGLM and ChatGPT.

with its [official code](#).

4.5 Main Results

Table 2 contains the experimental results of comparison between PRO and other baselines. To verify that PRO has a globally competitive ability to capture human preference from rankings with diverse lengths, we do experiments on *HH-RLHF*_{raw}, *HH-RLHF*_{Alpaca,3} and *HH-RLHF*_{Chatgpt,3}, last 2 of which are augmented from *HH-RLHF*_{raw} by Alpaca and ChatGPT, respectively.

In general, it can be found that LLaMA with fine-tuning has a notable improvement on BLEU and Reward against initial LLaMA, which has not undergone any specific alignment with human preference. Also, even without fine-tuning on *HH-RLHF*, models tuned on large-scale corpus still show certain performance, while ChatGLM and ChatGPT with RLHF training beat LLaMA, Curie, and Alpaca that are trained from scratch. All of these prove the significance of Human Alignment.

Next, we compare different human alignment algorithms using the same backbone on the same dataset. Even in the most basic setting of *HH-RLHF*_{raw}, where the ranking length is 2 with only one positive example and one negative example, PRO has already significantly outperformed all baselines in terms of reward score while maintaining considerable BLEU scores. Specifically, com-

pared to SFT, PRO improves the reward score by 6.52 points, and compared to the state-of-the-art human alignment algorithm RRHF, it improves the score by 3.1 points. This demonstrates that even without expanding the ranking sequence, PRO remains the best-performing approach. CoH achieves higher BLEU scores but falls short of PRO in terms of reward, which is mediocre. PRO exhibits a distinct advantage in terms of Harmlessness compared to Helpfulness. We attribute this to the fact that achieving Harmlessness is comparatively easier for PRO as it primarily involves significant features such as adapting expression styles and maintaining politeness in most conversations. On the other hand, **Helpfulness** typically demands more specific suggestions, which pose a greater challenge for language models due to their limited world knowledge, thus increasing the difficulty in this aspect.

When expanding the ranking sequence using existing LLMs and sorting it with an additional reward model (different from the evaluation reward model), we find that the utilized LLM plays a crucial role in achieving concrete performances. Since ChatGPT surpasses Alpaca in understanding human preferences, it provides superior samples during data augmentation compared to Alpaca, making *HH-RLHF*_{Chatgpt,i} intuitively better than *HH-RLHF*_{Alpaca,3}. The performance of each method increases from *HH-RLHF*_{Chatgpt,3} to *HH-*

*RLHF*_{Alpaca,3}. On the expanded sequences, we observe that BoN (selecting the response with the highest reward model score for SFT) becomes a competitive baseline. This finding aligns with Rafailov et al. 2023, who observed that RLHF is less tuning-efficient than BoN. The effectiveness of RRHF becomes less prominent because it relies on pairwise comparisons between candidates from given rankings. It fails to capture global differences corresponding to human preference in the long rankings, which can be achieved through Equation 5. Overall, in the expanded ranking, PRO remains the best-performing method, and the more powerful the LLM used for ranking augmentation, the more pronounced the improvement of PRO’s performance. This surprising characteristic fills us with anticipation for PRO’s future development.

4.6 Human and GPT-4 Evaluation

One could argue that the reward model fails to capture all human preferences in evaluation. Human annotation is considered the most accurate evaluation method, and recently, GPT-4-as-a-judge has emerged as a scalable approach for rapidly assessing human preference. Therefore, in this section, we provide comprehensive evaluations conducted by both GPT-4 and humans. To address cost concerns, we primarily compare the performance of PRO against two alternative counterparts:

PRO vs. Golden, i.e. the 1st candidate provided by the datasets, we aim to determine whether PRO trained on *HH-RLHF*_{raw} can achieve or surpass human-preferred responses provided by the raw dataset.

PRO vs. RRHF, both of which are trained on *HH-RLHF*_{raw}.

We aim to verify whether PRO is truly preferred over RRHF in terms of human preferences, even in ranking sequences of length 2 that do not fully exploit PRO’s capabilities. On the other hand, this comparison serves as evidence to some extent for the validity of the reward model we use in evaluation.

For GPT-4 evaluation, we first sample contexts in each test set. We assemble two corresponding responses from PRO and its counterparty into a modified version of the prompt template from Zheng et al. (2023) for GPT-4 scoring. We also refer to Wang et al. (2023) to provide two candidates in binary directions respectively, to eliminate unfairness triggered by candidate order. For Human evalua-

	Sub-set	Win	Tie	Lose
PRO vs. Golden	Harmless _{base}	60.00	5.00	35.00
	Helpful _{base}	77.50	0.00	22.50
	Helpful _{online}	27.50	12.50	60.00
	Helpful _{rejection}	55.00	0.00	45.00
	Average	55.00	4.37	40.63
PRO vs. RRHF	Harmless _{base}	62.50	10.00	27.50
	Helpful _{base}	70.00	0.00	30.00
	Helpful _{online}	45.00	5.00	50.00
	Helpful _{rejection}	65.00	0.00	35.00
	Average	60.62	3.75	35.63

Table 3: Results of GPT-4 Evaluation. We allow GPT-4 to evaluate responses between PRO and golden samples, as well as responses between PRO and RRHF.

	Sub-set	Win	Tie	Lose
PRO vs. Golden	Harmless _{base}	20.00	55.00	25.00
	Helpful _{base}	20.00	60.00	20.00
	Helpful _{online}	20.00	50.00	30.00
	Helpful _{rejection}	30.00	60.00	10.00
	Average	22.50	56.25	21.25
PRO vs. RRHF	Harmless _{base}	45.00	40.00	15.00
	Helpful _{base}	35.00	45.00	20.00
	Helpful _{online}	45.00	30.00	25.00
	Helpful _{rejection}	35.00	35.00	30.00
	Average	40.00	37.50	22.50

Table 4: Results of Human Evaluation.

tion, we employ human labelers to estimate the same samples with GPT-4 evaluation, and directly distinguish one response from another.

Table 3 and 4 give the detailed results, where both GPT-4 and Human more support PRO globally for each comparison, thus highlighting the strengths of PRO. We are surprised to find that both humans and GPT-4 consider the predictions of PRO to be better than the human-preferred responses annotated in the dataset. This suggests that PRO is able to effectively capture the preferences of humans as reflected in the annotated data. Furthermore, our evaluation using the reward model yielded consistent results, with both humans and GPT-4 significantly favoring PRO over RRHF. This not only reaffirms the effectiveness of PRO but also demonstrates that our reward model can reasonably evaluate human preferences.

4.7 Ablation Study

In this part, we investigate the effectiveness of each part in PRO, the results are contained in Table 5.

Traning set	Method	Harmless _{base}		Helpful _{base}		Helpful _{online}		Helpful _{rejection}		Total	
		BLEU	Reward	BLEU	Reward	BLEU	Reward	BLEU	Reward	BLEU	Reward
<i>HH-RLHF_{raw}</i>	PRO	12.05	62.96	20.83	48.51	28.75	59.02	27.17	53.28	21.54	55.35
	$-\mathcal{L}_{\text{SFT}}$	6.94	67.20	10.37	46.60	11.17	49.33	11.32	48.84	9.85	53.25
	$-\mathcal{T}$	12.04	62.91	20.63	47.92	28.73	58.52	26.94	53.08	21.41	55.04
	$-\mathcal{L}_{\text{SFT}} - \mathcal{T}$	0.88	52.81	6.74	42.97	6.37	42.84	6.85	44.71	5.14	46.17
<i>HH-RLHF_{Alpaca,3}</i>	PRO	14.41	62.6	22.47	54.38	25.61	60.90	26.82	58.26	22.11	58.72
	$-\mathcal{L}_{\text{PRO}}^{k>1}$	13.38	62.88	21.50	53.48	24.56	60.32	25.81	57.15	21.10	58.11
	$-\mathcal{L}_{\text{SFT}}$	9.06	65.78	18.77	54.18	23.90	62.26	23.33	58.29	18.29	59.71
	$-\mathcal{T}$	13.71	63.40	21.70	53.77	24.84	60.36	26.01	57.34	21.34	58.40
	$-\mathcal{L}_{\text{SFT}} - \mathcal{T}$	0.52	55.90	2.13	23.41	3.56	23.44	2.66	23.82	2.05	32.33
<i>HH-RLHF_{ChatGPT,3}</i>	PRO	15.53	73.08	22.30	64.78	29.35	66.66	27.49	66.95	23.07	67.97
	$-\mathcal{L}_{\text{PRO}}^{k>1}$	15.20	72.64	21.94	64.44	29.17	66.97	27.29	66.80	22.80	67.75
	$-\mathcal{L}_{\text{SFT}}$	13.81	73.18	21.28	64.20	27.90	67.15	26.57	66.76	21.84	67.84
	$-\mathcal{T}$	15.77	72.99	22.13	65.34	29.03	67.48	27.28	67.54	22.98	68.40
	$-\mathcal{L}_{\text{SFT}} - \mathcal{T}$	5.93	69.61	5.22	33.92	9.33	31.81	6.11	33.52	6.25	43.16

Table 5: Ablation results. We investigate the effectiveness of \mathcal{L}_{PRO} , \mathcal{L}_{SFT} and the dynamic temperature \mathcal{T} .

SFT Loss To avoid the model solely catering to the reward model at the expense of text quality, we introduce \mathcal{L}_{SFT} . Therefore, removing \mathcal{L}_{SFT} lowers BLEU scores on three datasets, but higher quality of corpus can, to some extent, compensates for its drop, as is proved by results on *HH-RLHF_{Alpaca,3}* and *HH-RLHF_{ChatGPT,3}* compared with *HH-RLHF_{raw}*.

PRO Loss Table 2 also demonstrates the influence of \mathcal{L}_{PRO} , as excluding it in PRO essentially equals to SFT (BoN) that gets lower Reward.

Adequate Ranking To fully leverage the ranking $y^1 \succ y^2 \succ \dots \succ y^n$, we employ $n - 1$ loss functions to model $y^1 \succ y^2, \dots, y^n, y^2 \succ y^3, \dots, y^n, \dots, y^{n-1} \succ y^n$. Our objective is to adequately model all ranking orders and enable LLM to better differentiate between samples of different preferences. To validate this idea, we deactivate \mathcal{L}_{PRO} except for its first term, $\mathcal{L}_{\text{PRO}}^1$. Experimental results on three datasets consistently demonstrate a decrease in both BLEU and Reward scores, thus confirming the effectiveness of Equation 5.

Temperature PRO With or without temperature (\mathcal{T}), the model performs well, but \mathcal{T} slightly enhances overall performance. Furthermore, we observe a significant drop in model performance when both the SFT loss and temperature are removed simultaneously, whereas removing either one individually did not have such a noticeable impact. We believe this is because temperature helps the model understand that some negative examples are neutral (with reward scores similar to positive examples), and thus should not be overly penalized

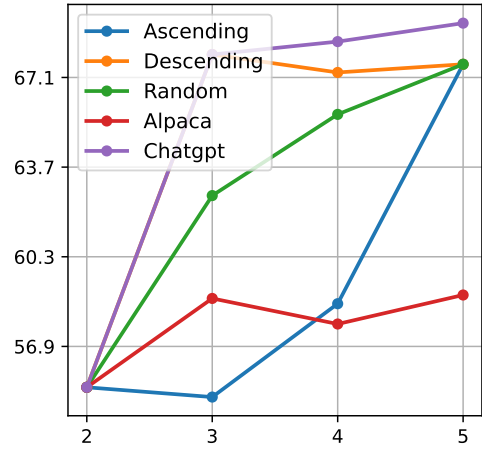


Figure 3: Results of experiments on different ranking lengths.

to avoid confusion during LLM training. Similarly, the inclusion of SFT loss also plays a similar role by increasing the weight of the best response.

4.8 Discussions

4.8.1 How about continually expanding Preference Ranking Sequence?

In Table 2, we have observed that expanding the ranking sequence of *HH-RLHF_{raw}* from length 2 to 3 using LLMs improves the performance of all models. This leads us to wonder how the effect would change if we further expand the preference ranking sequence. Specifically, we simulate 5 expansion strategies, each introducing 3 additional responses to extend the preference sequence to length 5, followed by reranking using a reward model.

Alpacas: Using Alpaca-7B, we generate 3 responses, adding 1, 2, and 3 responses, respectively,

Training set	Method	Harmless _{base}		Helpful _{base}		Helpful _{online}		Helpful _{rejection}		Total	
		BLEU	Reward	BLEU	Reward	BLEU	Reward	BLEU	Reward	BLEU	Reward
<i>HH-RLHF_{raw}</i>	PRO	12.05	62.96	20.83	48.51	28.75	59.02	27.17	53.28	21.54	55.35
	PRO _s	16.84	59.27	22.34	48.22	30.13	58.23	28.21	53.41	23.77	54.20
<i>HH-RLHF_{Alpaca,3}</i>	PRO	14.41	62.6	22.47	54.38	25.61	60.90	26.82	58.26	22.11	58.72
	PRO _s	13.44	62.44	21.18	52.82	23.01	59.07	25.36	56.51	20.68	57.44
<i>HH-RLHF_{ChatGPT,3}</i>	PRO	15.53	73.08	22.30	64.78	29.35	66.66	27.49	66.95	23.07	67.97
	PRO _s	15.53	73.16	22.02	65.34	29.04	67.18	27.49	67.41	22.96	68.36

Table 6: Results of diverse self-bootstrapping policies.

to form ranking sequences of lengths 3, 4, and 5.

ChatGPT: Using ChatGPT, we generate three responses, adding 1, 2, and 3 responses, respectively, to form ranking sequences of lengths 3, 4, and 5.

Ascending: We utilize three LLMs, namely Curie, Alpaca-7B, and ChatGPT. Based on the zero-shot results in Table 2, the quality of their responses can be ranked as ChatGPT > Alpaca-7B > Curie. In the Ascending setting, we add the responses in ascending order of quality. That is, for a sequence of length 3, we added Curie’s response; for a sequence of length 4, we added Curie and Alpaca-7B’s responses; and for a sequence of length 5, we added Curie, Alpaca-7B, and ChatGPT’s responses.

Descending: The data source is the same as Ascending, but the responses are added in the opposite order. For a sequence of length 3, we added ChatGPT’s response; for a sequence of length 4, we added ChatGPT and Alpaca-7B’s responses; and for a sequence of length 5, we added Curie, Alpaca-7B, and ChatGPT’s responses.

Random: The order of response additions is unrelated to response quality and is done randomly.

In Figure 3, we present the impact of various expansion strategies on the effectiveness of PRO after expanding sequences of different lengths. Our observations are as follows:

Longer Ranking, Better results: Overall, longer ranking sequences generally lead to improved performance for most strategies, which is an exciting finding, as expanding the ranking sequence is a relatively straightforward task compared to designing new prompts.

Better added responses, better results: If a single model is used to generate additional responses, supplementing one response is sufficient when the quality is average, such as with Alpaca, adding more responses provides limited improvement. However, when the quality of responses is high, as with ChatGPT, adding more responses leads to consistent performance gains. This could

potentially offer new insights for the design of future Human Alignment algorithms.

More diversified added responses, better results: We have also discovered that incorporating lower-quality responses may actually improve the model’s results compared to using only high-quality responses. Interestingly, when the sequence length is 4, Ascending (blue line) =Curie+Alpaca surpasses the performance of Alpaca(red line)=Alpaca+Alpaca, even though Curie’s response quality is not as good as Alpaca’s. We believe this is because diverse responses, even if they are negative examples, help the language model become more aware of behaviors that should be avoided, thereby enhancing overall performance. Lastly, by combining Curie, Alpaca, and ChatGPT, we achieve a performance close to using three ChatGPT responses, demonstrating the truth in the saying, "Two heads are better than one."

4.8.2 Can self-bootstrapping augmentation enhance performance?

We have demonstrated the effectiveness of incorporating responses from other LLMs to expand ranking length, which significantly improves human preference. A natural question arises: Can we further improve the model’s performance by including responses from the LLM itself in the candidate list? This can be seen as a special approach to expanding preference ranking sequences.

From Table 6, we find that self-bootstrapping⁶ exhibits conflicting results. On *HH-RLHF_{raw}*, self-bootstrapping shows an improvement in BLEU but a slight decrease in reward score. On *HH-RLHF_{Alpaca,3}*, both BLEU and reward score decrease. However, on *HH-RLHF_{ChatGPT,3}*, self-bootstrapping improves reward score while maintaining BLEU value. We speculate that self-

⁶The naive self-bootstrapping makes LLMs easily overfit RM_{train} . We accordingly regularize it by preventing the augmented candidate from taking the position of the originally top 1, and re-ranking all reward to ensure the descending order.

bootstrapping is effective only when the underlying language model is strong. Furthermore, although self-bootstrapping enhances performance on $HH\text{-}RLHF_{\text{ChatGPT},3}$, it can be seen as extending the ranking sequence to 4, and the improvement may not be as significant as adding an additional high-quality response generated by ChatGPT. We also acknowledge that these relatively negative results may stem from training a 7B model with a reward model of size 1.4B. Expanding the model size might yield more exciting performance gains, similar to the scaling law of RLHF (Ouyang et al., 2022b; Gao et al., 2022), which we leave for future work.

5 Related Work

5.1 Reinforcement Learning from Human Feedback

Fine-tuning language models to align with human preferences has emerged as a critical research problem. It can be formulated as given a context and corresponding suffixes ranked or scored by human annotators without more detailed labels, the agent is required to learn human preference and provide human-like results. Reinforcement Learning (RL) plays the most straightforward way to reach this goal, for the agent needs just scarce supervision signal from reward models as human proxies, and is modified through numerous trials under RL framework, namely **Reinforcement Learning from Human Feedback** (RLHF). Many explorations have been done on this path (Christiano et al., 2017; MacGlashan et al., 2017; Warnell et al., 2018; Ziegler et al., 2019; Stiennon et al., 2020b; Nakano et al., 2021; Lee et al., 2021; Lei et al., 2022; Snell et al., 2022; Bai et al., 2022a; Ouyang et al., 2022a). Lei et al. (2022) implement an online RL scenario by establishing a user simulator commonly for training and evaluation, which at each session is initialized with different Gaussian vectors representing diverse personalities. In contrast, ILQL is applicable for the offline setting, which is released by Snell et al. (2022). Stiennon et al. (2020b) and Nakano et al. (2021) investigate the RLHF method for text summarization and question answering, respectively. Bai et al. (2022a) apply RLHF to enable LLMs to become harmless and helpful, while releasing a new conversational dataset with human feedback. Known as a masterpiece, Ouyang et al. (2022a) propose InstructGPT which is first fine-tuned in a supervised way, then continually modified under PPO algorithm (Schul-

man et al., 2017). This process is cyclic, during which the performance of the trained agent spirals upwards. It is also applied to the famous ChatGPT by OpenAI.

5.2 Supervised Fine-tuning for Human Preference Alignment

Despite appealing advantages, RL-based methods have obvious limitations regarding training efficiency and complexity, consequently driving researchers to focus on Supervised Fine-tuning methods without these challenges. Liu et al. (2023) combine desirable and undesirable suffixes in a template prompted by opposite keywords, thus fully dependent on a highly semantic understanding of large language models. Yuan et al. (2023) compose multiple pairwise comparisons between suffixes in the given ranking, which forms a new algorithm from the perspective of training objectives. Rafailov et al. (2023) similarly transform LLMs as a Bradley-Terry model to measure chosen and rejected candidates by human annotators. The proposed PRO chooses the path of modifying the SFT objective, but is further promoted from RLHF formulation and inherits its straightforwardness towards Human Preference Alignment. In particular, PRO transforms RL’s indirect optimization into a direct one, and extends pairwise comparisons to multi-dimensional and multi-positional comparisons. Comprehensive experiments prove its excellence in human preference acquisition while maintaining the quality of generated texts.

6 Conclusion

In this paper, we derive from the Bradley-Terry comparison of the reward model in RLHF that human alignment can be modeled as aligning the probability ranking of n responses generated by the LLM and the preference ranking of these responses by humans. Based on this derivation, we propose PRO. PRO inherits the advantages of RLHF, and further captures fine-grained distinction corresponding to human preference from multiple one-to-many comparisons. We conduct extensive experiments to verify the excellence of PRO against other baselines and investigate the impact of multifaceted factors. Overall, the findings presented in this paper demonstrate the significance of PRO in effectively and efficiently aligning LLMs to human preference. This work can serve as a stepping stone for further quantifiable explorations.

Disclaimer

Since some services provided by OpenAI are currently not available in mainland China, data augmentation and inference from ChatGPT, as well as GPT-4 evaluation, are completed where the related services are available.

There exists sensitive and offensive content in HH-RLHF, which aims for only research purposes. Viewpoints included in the data do not represent our attitudes. We hope our work can be used to make AI technologies in line with ethical requirements.

References

- Afra Feyza Akyürek, Ekin Akyürek, Aman Madaan, Ashwin Kalyan, Peter Clark, Derry Wijaya, and Niket Tandon. 2023. [R14f: Generating natural language feedback with reinforcement learning for repairing model outputs](#). *arXiv preprint arXiv:2305.08844*.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022a. [Training a helpful and harmless assistant with reinforcement learning from human feedback](#). *arXiv preprint arXiv:2204.05862*.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022b. [Constitutional ai: Harmlessness from ai feedback](#). *arXiv preprint arXiv:2212.08073*.
- 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.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020b. [Language models are few-shot learners](#). *Advances in neural information processing systems*, 33:1877–1901.
- Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrmann, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. [Sparks of artificial general intelligence: Early experiments with gpt-4](#). *arXiv preprint arXiv:2303.12712*.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. [Palm: Scaling language modeling with pathways](#). *arXiv preprint arXiv:2204.02311*.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martić, Shane Legg, and Dario Amodei. 2017. [Deep reinforcement learning from human preferences](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. [GLM: General language model pretraining with autoregressive blank infilling](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, Dublin, Ireland. Association for Computational Linguistics.
- Leo Gao, John Schulman, and Jacob Hilton. 2022. [Scaling laws for reward model overoptimization](#). *arXiv preprint arXiv:2210.10760*.
- Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. 2023. [Reward design with language models](#). *arXiv preprint arXiv:2303.00001*.
- Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, et al. 2022. [The bigscience roots corpus: A 1.6 tb composite multilingual dataset](#). *Advances in Neural Information Processing Systems*, 35:31809–31826.
- Kimin Lee, Laura Smith, and Pieter Abbeel. 2021. [Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training](#). *arXiv preprint arXiv:2106.05091*.
- Wenqiang Lei, Yao Zhang, Feifan Song, Hongru Liang, Jiaxin Mao, Jiancheng Lv, Zhenglu Yang, and Tat-Seng Chua. 2022. [Interacting with non-cooperative user: A new paradigm for proactive dialogue policy](#). In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '22*, page 212–222, New York, NY, USA. Association for Computing Machinery.
- Minghao Li, Feifan Song, Bowen Yu, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. 2023. [Api-bank: A benchmark for tool-augmented llms](#). *arXiv preprint arXiv:2304.08244*.

- Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. 2023. [Chain of hindsight aligns language models with feedback](#). *arXiv preprint arXiv:2302.02676*.
- James MacGlashan, Mark K. Ho, Robert Loftin, Bei Peng, Guan Wang, David L. Roberts, Matthew E. Taylor, and Michael L. Littman. 2017. [Interactive learning from policy-dependent human feedback](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2285–2294. PMLR.
- Niklas Muennighoff, Alexander M Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. 2023. [Scaling data-constrained language models](#). *arXiv preprint arXiv:2305.16264*.
- Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. [Webgpt: Browser-assisted question-answering with human feedback](#). *arXiv preprint arXiv:2112.09332*.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. [Representation learning with contrastive predictive coding](#). *arXiv preprint arXiv:1807.03748*.
- OpenAI. 2023. [Gpt-4 technical report](#). *arXiv preprint arXiv:2303.08774*.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll 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 F Christiano, Jan Leike, and Ryan Lowe. 2022a. [Training language models to follow instructions with human feedback](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022b. [Training language models to follow instructions with human feedback](#). *Advances in Neural Information Processing Systems*, 35:27730–27744.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. [Instruction tuning with gpt-4](#). *arXiv preprint arXiv:2304.03277*.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#). *arXiv preprint arXiv:2305.18290*.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal policy optimization algorithms](#). *arXiv preprint arXiv:1707.06347*.
- Charlie Snell, Ilya Kostrikov, Yi Su, Mengjiao Yang, and Sergey Levine. 2022. [Offline rl for natural language generation with implicit language q learning](#). *arXiv preprint arXiv:2206.11871*.
- Kihyuk Sohn. 2016. [Improved deep metric learning with multi-class n-pair loss objective](#). *Advances in neural information processing systems*, 29.
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020a. [Learning to summarize with human feedback](#). *Advances in Neural Information Processing Systems*, 33:3008–3021.
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020b. [Learning to summarize with human feedback](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. [Llama: Open and efficient foundation language models](#). *arXiv preprint arXiv:2302.13971*.
- Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023. [Large language models are not fair evaluators](#). *arXiv preprint arXiv:2305.17926*.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hananeh Hajishirzi. 2022. [Self-instruct: Aligning language model with self generated instructions](#). *arXiv preprint arXiv:2212.10560*.
- Garrett Warnell, Nicholas Waytowich, Vernon Lawhern, and Peter Stone. 2018. [Deep tamer: Interactive agent shaping in high-dimensional state spaces](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen,

- Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Zequi Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. [Fine-grained human feedback gives better rewards for language model training](#). *arXiv preprint arXiv:2306.01693*.
- Wanqi Xue, Bo An, Shuicheng Yan, and Zhongwen Xu. 2023. [Reinforcement learning from diverse human preferences](#). *arXiv preprint arXiv:2301.11774*.
- 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*.
- Tianjun Zhang, Fangchen Liu, Justin Wong, Pieter Abbeel, and Joseph E Gonzalez. 2023. [The wisdom of hindsight makes language models better instruction followers](#). *arXiv preprint arXiv:2302.05206*.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. [Judging llm-as-a-judge with mt-bench and chatbot arena](#). *arXiv preprint arXiv:2306.05685*.
- Chunting Zhou, Pengfei Liu, Puxin Xu, Srinu Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. [Lima: Less is more for alignment](#). *arXiv preprint arXiv:2305.11206*.
- Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. [Fine-tuning language models from human preferences](#). *arXiv preprint arXiv:1909.08593*.