Data with High and Consistent Preference Difference Are Better for Reward Model

Qi Lin[1], **Hengtong Lu**[1], **Caixia Yuan**[1], **Xiaojie Wang**,[1*], **Huixing Jiang**,[2 *], **Wei Chen**[2]
[1]Beijing University of Posts and Telecommunications Beijing, China
[2]Li Auto Inc. Beijing, China
{07_2018, luhengtong, yuancx, xjwang}@bupt.edu.cn
{jianghuixing, chenwei10}@lixiang.com

Abstract

Reinforcement Learning from Human Feedback (RLHF) is a commonly used alignment method for Large Language Models (LLMs). This method relies on a reward model trained on a preference dataset to provide scalar rewards. However, the human-annotated preference data is often sparse, noisy, and costly to obtain, necessitating more efficient utilization. This paper proposes a new metric for better preference data utilization from both theoretical and empirical perspectives. Starting with the Bradley-Terry model, we compute the Mean Square Error (MSE) between the expected loss and empirical loss of the reward model. Our findings reveal that data with higher and more consistent difference result in lower MSE. We therefore propose the Preference Difference (PD), the reward difference between two samples, as a filter for preference data. Experimental results on three open-source models show that reward models trained by filtered data with PD achieve higher calibrated accuracy, as well as better RLHF alignment performance. The conclusion remains consistent when we extend the experiments and theoretical derivations to implicit reward alignment algorithms, such as Direct Preference Optimization (DPO).

Code, dataset and extended version — github.com/07v2018/Data-with-High-and-Consistent-Preference-Difference-Are-Better-for-Reward-Model

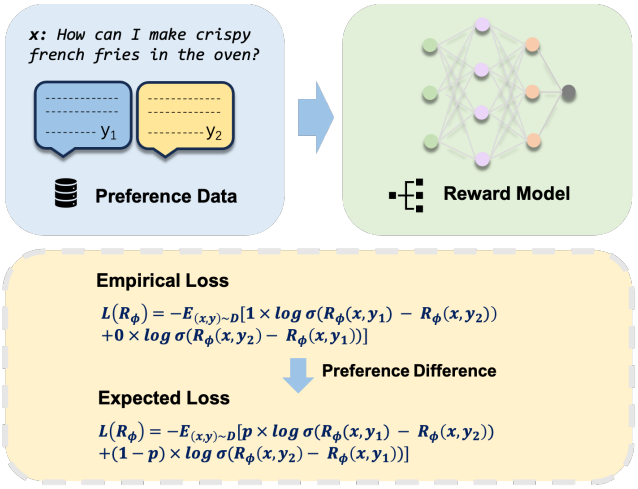

Figure 1: Due to the labeling method of preference data, there is a gap between the empirical loss and the expected loss in reward model training. Using the PD metric to filter the data can effectively narrow the gap between empirical loss and expected loss.

Introduction

Large language models (LLM) have achieved unprecedented success in a broad range of natural language processing (NLP) tasks, such as question-answering, summarization, and dialogue (Brown et al. 2020). However, they may not align well with human values such as harmlessness, helpfulness, and honesty (HHH). Reinforcement Learning from Human Feedback (RLHF) (Christiano et al. 2017; Stiennon et al. 2020) was proposed to handle the problem to achieve value alignments between human and LLM outputs. While RLHF has been shown to be effective, but many problems are still to be addressed for a more stable and efficient RLHF process.

The existing RLHF pipeline heavily depends on reward models which are trained on a set of paired samples with preference labels. Lots of attention has been paid to building better reward models. Some previous works design elaborate loss functions to make better use of the preference data, such as adding a margin in the original loss function or a difficulty decay coefficient to the ranking loss in the loss function(Cai et al. 2024; Touvron et al. 2023a). However these works only try to improve the performance of reward models by utilizing the loss function, without delving into the preference data itself. The way in which reward models depend on the characteristics of preference data remains unclear.

This paper tries to address the problem from both theoretical and experimental aspects. We demonstrated that preference data with higher differences is more effective than that with lower differences, and data with more consistent differences is superior to those with diverse differences for training reward models. We point out that due to the low consistency found in human annotators in previous work (Bai et al. 2022a; Dubois et al. 2024), the binary labels used in most preference datasets (Bai et al. 2022a; Stiennon et al.

*Corresponding author

2020; Cui et al. 2024) can lead to weight bias during reward model training.

Starting from the Bradley-Terry model (Bradley and Terry 1952), we first give both the empirical and expected loss function for training the reward model using human preference data. Then we employ the Mean Squared Error (MSE) to measure the difference between two loss functions. We find that preference data with higher and more consistent data differences can reduce the MSE between two loss functions, thereby improving the performance of the reward model. Based on the findings, we propose a Preference Difference (PD) metric which provides a simple way to select preference data with higher and more consistent data differences, thereby reducing the gap between expected loss and empirical loss, as illustrated in Figure 1. Experimental results on three different open-source models show that reward models trained on selected data by PD achieve better RLHF alignment performance. It is also effective for implicit reward alignment algorithms like DPO (Rafailov et al. 2023).

In summary, our contributions are as follows:

- We theoretically show that data with high mean and low variance has a smaller MSE between the currently used empirical loss function and the expected loss function of the reward model, which helps train a better reward model.
- We propose a new metric, named Preference Difference (PD). The metric is easy to calculate, and efficient to select the data with high-difference and high-consistency.
- Experimental results show that training with data selected by PD improves the calibrated accuracy of the reward model, as well as alignment performance. It is also effective for implicit reward alignment algorithms like DPO.

Related Work

Reinforcement Learning from Human Feedback

Nowadays most LLMs are built on pre-trained language models, which leads to a gap with downstream tasks. Aligning LLMs helps us bridge this gap, ensuring that the language generated is coherent, contextually accurate, and aligns with what humans intend to convey. To align with human values, pre-trained language models are then trained through interaction with humans (Ouyang et al. 2022; OpenAI 2023), with a combination of supervised learning and reinforcement learning. Incorporating human feedback into the process of language model (LM) training has been shown to be effective in reducing helpless, harmful, and other undesired model generation outputs. There have been many attempts on this path recently (OpenAI 2023; Bai et al. 2022b,a; Ziegler et al. 2019).

Reinforcement learning from human feedback (RLHF) has been proven to be an advanced way to align LMs with human values (Bai et al. 2022a). The process of RLHF begins with the collection of human feedback, which can come in various forms, such as explicit ratings, preference comparisons, or qualitative comments (Ziegler et al. 2019; Stiennon et al. 2020; Bai et al. 2022a). This feedback is then used to train a reward model that encapsulates the preferences and values of the human annotators. The reward model serves as a guide for the subsequent reinforcement learning process, where the language model is optimized to generate outputs that align with these human preferences. However, successful RLHF training requires a lot: an accurate reward model as a surrogate for human judgment, careful hyperparameter exploration for stable parameter updating, and a strong PPO algorithm for robust policy optimization (Schulman et al. 2017). The reward model trained on low-quality data and hard-to-define alignment target can easily mislead the PPO algorithm to an unintelligible direction (Zheng et al. 2023; Bai et al. 2022a). Our study analyzed what constitutes higher-quality reward data and investigated whether reward data should emphasize diversity or consistency in preference difference levels. In contrast to many studies (Touvron et al. 2023a; Wang et al. 2024) that use margin to amplify the reward model's differentiation between positive and negative samples, we emphasize the core aspect of expanding the diversity within the data itself.

Reward Model and Preference Data

The reward model plays a central role in RLHF, which directly converts human preference into a reward signal (Ouyang et al. 2022; Bai et al. 2022a). Due to the instability reflected by humans directly labeling scores, preference datasets are often paired or grouped data with ranking results (Christiano et al. 2017; Ouyang et al. 2022). Specifically, humans are presented with two or more outputs and asked to select one or rank them, and this result is then used to train a reward model (Stiennon et al. 2020). Some efforts are made to investigate the relationship between preference data and the reward model. Prior works (Gao, Schulman, and Hilton 2023) investigate the relationships between the size of the reward model dataset, the number of reward and policy models' parameters, and the coefficient of the KL penalty added to the reward in the reinforcement learning setup. Anthropic (Bai et al. 2022a) discussed the scaling results of model parameters and the sizes of the dataset, releasing the HH-RLHF dataset. The aforementioned work, along with some efforts to modify the reward model loss function (Bai et al. 2022a; Dubois et al. 2024), superficially addresses the impact of difference of preference data on the reward model but lacks in-depth exploration. Our work focuses on the impact of the data difference within the pair-wised preference samples which is not yet addressed in previous works. Based on our conclusions, we further propose a simple and efficient data filtering method to improve the alignment performance of RLHF.

Preliminaries

Reward Model for RL Fine-tuning

Let $(y_1, y_2|x)$ be a pair-wised preference sample, where y_1, y_2 denotes two responses to the query x, and y_1 is more preferred by the human annotators. The reward model is denoted as $R_\phi(x, y)$. Following the previous work for human preference modeling (Sadigh et al. 2017; Bai et al. 2022a) and Bradley-Terry model (Bradley and Terry 1952), with the

preference dataset denoted as $D = \{x^i, y_1^i, y_2^i\}_{i=1}^N$, the preference probability for reward function $R_\phi(x, y)$ is outlined as in Equation 2:

$$P_\phi(y_1 \succ y_2|x) = \frac{\exp\big(R_\phi(x, y_1)\big)}{\exp\big(R_\phi(x, y_1)\big) + \exp\big(R_\phi(x, y_2)\big)} = \sigma\big(R_\phi(x, y_1) - R_\phi(x, y_2)\big) \quad (1)$$

where σ is the logistic sigmoid function. Then the loss function used for training reward models can be expressed as in Equation 2:

$$\begin{aligned}\mathcal{L}(R_\phi) &= -\mathbb{E}_{(x,y_1,y_2)\sim D}[(\log\sigma\big(R_\phi(x, y_1) - R_\phi(x, y_2)\big))] \\ &= -\mathbb{E}_{(x,y_1,y_2)\sim D}[1 \times \log\sigma\big(R_\phi(x, y_1) - R_\phi(x, y_2)\big) \\ &\quad +0 \times \log\sigma\big(R_\phi(x, y_2) - R_\phi(x, y_1)\big)]\end{aligned} \quad (2)$$

The trained reward model is then used for the RL fine-tuning process on the language model. The language model responds to the input prompts and outputs the corresponding responses, the reward model provides reward values for the language model's responses.

Direct Preference Optimization

Besides RL fine-tuning, Direct Preference Optimization (DPO) is an implicit reward alignment method that also uses preference datasets to directly optimize language models. Since it has been widely applied in language model alignment, we also take it into our consideration.

In the DPO algorithm (Rafailov et al. 2023), the optimal policy π^* under the Bradley-Terry model satisfies the preference model:

$$P_{\text{DPO}}(y_1 \succ y_2|x) = \sigma\left(\beta\log\frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)} - \beta\log\frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)}\right) \quad (3)$$

The loss function can be expressed as:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta) = -\mathbb{E}_{(x,y_1,y_2)\sim D}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_1|x)}{\pi_{\text{ref}}(y_1|x)} - \beta\log\frac{\pi_\theta(y_2|x)}{\pi_{\text{ref}}(y_2|x)}\right)\right] \quad (4)$$

Preference Difference

In this section, we first derive the expected and empirical loss function of the reward model. We then calculate the Mean Squared Error (MSE) between the two loss functions and find that data with higher and more consistent differences can reduce the bias. According to the findings, We finally propose a PD metric to select data with higher and more consistent data differences.

Loss Function of Reward Models

The commonly used preference datasets often have binary labels (Bai et al. 2022a; Dai et al. 2023; Stiennon et al. 2020). However, previous studies have shown that human annotators can be inconsistent, meaning different annotators might not always label the same sample in the same way (Bai et al. 2022a; Dubois et al. 2024). Generally, the BT

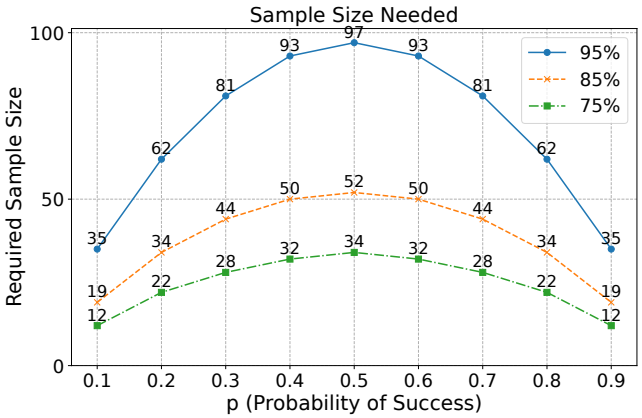

Figure 2: This chart illustrates the required binomial distribution sample times for different success probabilities p (ranging from 0.1 to 0.9) at confidence levels of 95%, 85%, and 75%, with a 10% error margin. It demonstrates that if we aim to sufficiently annotate each sample in the dataset to minimize error, the annotation cost would become unacceptable.

model provides a win-lose probability $P_\phi(y_1^i \succ y_2^i|x^i)$ as shown in Equation 2 for each sample pair $\{x^i, y_1^i, y_2^i\}$. If the annotation repeats T times, the process can be modeled as a binomial distribution Binomial$(T, P_\phi(y_1^i \succ y_2^i|x^i))$. Therefore the expected loss function for the reward model should be as in Equation 5:

$$\begin{aligned}\mathcal{L}_{\text{expect}}(R_\phi) = &-\mathbb{E}_{(x,y_1,y_2)\sim D} \\ &\big[p \times \log\sigma\big(R_\phi(x, y_1) - R_\phi(x, y_2)\big) \\ &+(1-p) \times \log\sigma\big(R_\phi(x, y_2) - R_\phi(x, y_1)\big)\big]\end{aligned} \quad (5)$$

where $p = P_\phi(y_1 \succ y_2|x)$.

While in reward model training, we actually employ the following empirical loss function as shown in Equation 6.

$$\begin{aligned}\mathcal{L}_{\text{empirical}}(R_\phi) = &-\mathbb{E}_{(x,y_1,y_2)\sim D} \\ &\big[w_1 \times \log\sigma\big(R_\phi(x, y_1) - R_\phi(x, y_2)\big) \\ &+w_2 \times \log\sigma\big(R_\phi(x, y_2) - R_\phi(x, y_1)\big)\big]\end{aligned} \quad (6)$$

where w_1, w_2 are the number of wins for y_1 and y_2 respectively.

In current open-source datasets, the preference label y_1, y_2 is usually a binary classification label, and the data pair $\{x^i, y_1^i, y_2^i\}$ appears only once, which means that $\hat{y} \sim$ Binomial$(1, P_\phi(y_1^i \succ y_2^i|x^i)), w_1 + w_2 = 1$. Equation 6 reduces to Equation 2 in this case.

Bias Analysis

In Equation 5, different samples $\{x^i, y_1^i, y_2^i\}$ will receive different weights as $P_\phi(y_1^i \succ y_2^i|x^i)$ varies. But in Equation 2 they share the same weights, which leads to a weight bias. We use MSE to measure the bias between the empirical loss $\mathcal{L}(R_\phi)$and the expected loss $\mathcal{L}_{\text{expect}}(R_\phi)$.

With $\sigma = \sigma\big(R_\phi(x, y_1) - R_\phi(x, y_2)\big), 1 - \sigma = \sigma\big(R_\phi(x, y_2) - R_\phi(x, y_1)\big)$:

$$\textbf{MSE} = \mathbb{E}_{(x,y_1,y_2)\sim D} \Big[\big(p \times \log\ \sigma + (1-p) \times \log(1-\sigma) - \log\ \sigma\big)^2\Big] \tag{7}$$

We finally get the following result shown in Equation 8, derivation details are listed in the appendix:

$$\textbf{MSE} = \Big[\text{Var}(p) + \big(\mathbb{E}_{(x,y_1,y_2)\sim D}(p) - 1\big)^2\Big] \times \mathbb{E}_{(x,y_1,y_2)\sim D}\left[\log\left(\frac{\sigma}{1-\sigma}\right)^2\right] \tag{8}$$

Due to $p = P_\phi(y_1 \succ y_2|x) \geq 0.5$, it implies that the MSE between the expected loss function $\mathcal{L}_{expect}(R_\phi)$ and the loss function $\mathcal{L}(R_\phi)$ is proportional to the preference consistency $Var(p)$ of the sample data and inversely proportional to the overall preference intensity $\mathbb{E}(p)$. The higher the preference consistency of the sample as a whole, the lower the MSE. At the same time, the higher the preference intensity, the lower the MSE.

The above analysis is based on the assumption that each sample only appears once. So can we approach the expected p by increasing the number of annotations for each sample in the preference data? Figure 2 shows the number of annotations required per single sample to achieve confidence levels of 75%, 85%, and 95%, with a 10% error margin, for different values of p. When p is between 0.3 and 0.7, the required number of annotations ranges from 28 to 97, resulting in an unacceptable workload for annotation. This necessitates finding ways to reduce the error by improving data quality.

Definition and Application of PD

According to the analysis in the above subsection, if the data used for training the reward model has bigger enough $\mathbb{E}(p)$ and smaller enough $Var(p)$, we can approximate expected loss with empirical loss. To achieve this, we define Preference Difference (PD) on data as in Equation 9:

$$\textbf{PD}_{R_\phi}(i) = |R_\phi(x^i, y_1^i) - R_\phi(x^i, y_2^i)| \tag{9}$$

The above formula describes the method for estimating PD using the reward values $R_\phi(x, y)$ calculated by the reward model in the RLHF pipeline. As for DPO, which utilizes implicit reward alignment and can similarly calculate the sample BT probability. Thereby, replacing $R_\phi(x, y)$ in the reward model of Equation 5,6,8 into $\beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$, we can draw similar conclusions from the DPO loss, with detailed derivation provided in the appendix.

Here we define $\textbf{PD}_{\text{DPO}}$ as below:

$$\textbf{PD}_{\text{DPO}}(i) = |\log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)} - \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)}| \tag{10}$$

Reminding the estimation of p in Equation 2, we have:

$$p^i = \sigma(R_\phi(x^i, y_1^i) - R_\phi(x^i, y_2^i)) \tag{11}$$

PD concerns the difference of the reward values between each sample pair. Sample pairs with smaller differences have smaller PD values, those with larger differences will have larger PD values.

We can find that $\textbf{PD}(i)$ is proportional to p^i. Therefore, data with large PD will increase the $\mathbb{E}(p)$, and data with consistent PD will reduce $Var(p)$. i.e. data with large and consistent PD have small MSE. It gives us a way for data selection to train reward models with less loss bias.

To obtain PD, we train an initial reward model with the empirical loss (Equation 2). With this initial reward model, we can calculate the PD value for each sample $\{x^i, y_1^i, y_2^i\}$ of preference data. By sorting the data samples according to their PD and then filtering out data according to a threshold for PD. Although the total amount of data decreases after filtering, we found that the performance improvement brought by this data filtering strategy can counteract or even surpass the impact of the reduced data volume in experiments.

Experiments

In this section, we evaluate how PD estimate the win-lose probability p. Then we show by experiment that by filtering data with PD, the reward model receives higher calibrated accuracy and the RLHF model achieves higher performance. And we discussed PD's influence on AI risk and bias.

Experiment Settings

Task and Model. We choose the safety alignment task to conduct our experiment (Ji et al. 2023). It aims to enhance helpfulness while reducing the harmfulness of the model output y when given a prompt x. To test models with different abilities, we choose models from Huggingface/open_llm_leaderboard (Beeching et al. 2023). Considering the average scores of models, we chose mistralai/Mistral-7B-v0.1 (Jiang et al. 2023), meta-Llama/Llama-2-7b-hf (Touvron et al. 2023b) and openlm-research/open_Llama_3b_v2 (Geng and Liu 2023), which respectively have average leaderboard scores of approximately 60, 50, and 40.

SFT phase. For the SFT phase, we used the Alpaca dataset as BeaverTails (Taori et al. 2023) for instruction fine-tuning. Alpaca is a dataset of 52k instructions and demonstrations generated by OpenAI's text-davinci-003 engine. Our SFT implementation and settings follow the implementation of safe-RLHF (Dai et al. 2023).

RLHF phase. For the reward model training phase and RL fine-tuning phase, we adopted the HH-RLHF dataset (Bai et al. 2022a). HH-RLHF is a dataset captured from real human feedback, ranked by human annotators both helpfully and harmlessly. It contains 160k pairs of data, consisting of 118k helpful and 42k harmless instances as the training set. Each sample consists of a dialogue history between a human and an assistant, followed by two subsequent responses. We referred to the data partitioning method used by MOSS (Zheng et al. 2023). We randomly sampled 12k harmless data and 28k helpful data for the RL phase and the rest of the data is used for reward model training. From the remaining 8.5k test data, we randomly selected 0.7k helpful and 0.3k harmless examples for a total of 1k data as our test set.

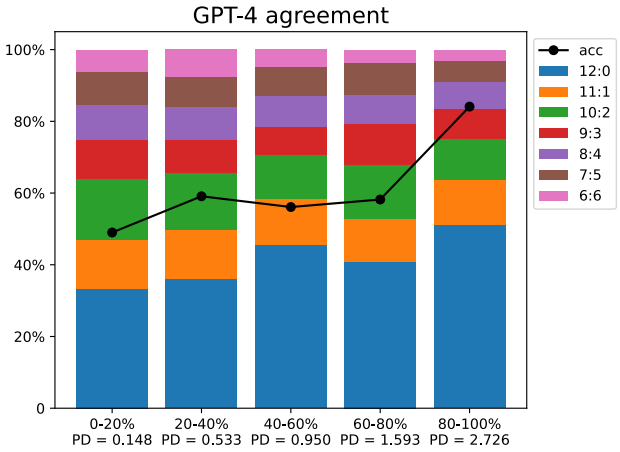

Figure 3: GPT-4 agreement for 512 samples in HH-RLHF train set. The results were inferred 12 times using GPT-4 and then divided into 5 equal parts based on PD. Different colors represent the proportion of data with varying degrees of consistency, and the average PD was represented below. The black line indicates the accuracy of GPT-4 in relation to the original dataset labels for both individuals, representing the consistency between the original dataset and GPT-4.

Evaluation. Our evaluation strategy is following DPO (Rafailov et al. 2023) and MOSS (Zheng et al. 2023), which uses GPT-4 as the evaluator. In detail, we collect the inference results for models under each of the following settings. Then we compared each model's win rate against the chosen response in test sets of the preference dataset. Higher win rates mean higher performance.

All the experiments are conducted on identically implemented machines. Each machine contains eight 80G A100 or A800 GPUs. We carry out full-parameter fine-tuning using the PPO implementation of safe-RLHF(Dai et al. 2023) .To save on GPU memory cost, we use DeepSpeed ZERO3 (Rasley et al. 2020), BF-16, and gradient checkpoint. More details about settings are available in the appendix.

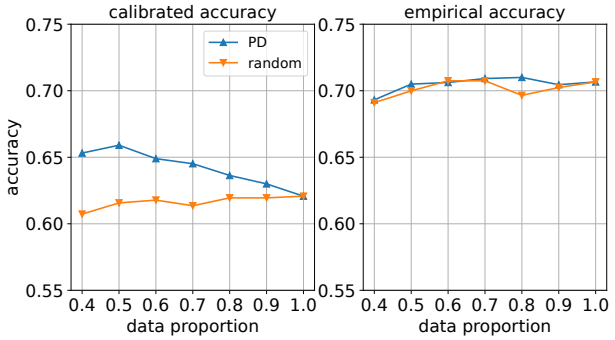

Figure 4: **Left part**: calibrated accuracy prediction for different Llama-2 reward models. **Right part**: actual accuracy for different Llama-2 reward models.

Evaluation for PD Metric

In this section, we evaluated the PD metric to see if it can really estimate p for data samples. Since p^i is the probability for Binomial(T, p^i), samples with higher p^i will show higher agreement in multiple annotations, the same as PD. So we use GPT-4 to re-annotate a subset of the original dataset. We sampled 512 samples from the original train dataset of HH-RLHF. Then we used the default reward model trained on the full dataset to compute PD for each sample. We requested GPT-4 to provide a comparison both helpfully and harmlessly for two responses based on each sample's prompt, repeating 12 times. (The prompt for GPT-4 can be found in the appendix.) We evaluated GPT-4's agreement and overall accuracy across four judgments. Each sample could result in one of 12 possible outcomes, from 12:0 to 6:6. These scores represent the preference for each of the two responses in the four judgments. The experiment results can be seen in Figure 3. We sorted the samples with their PD from low to high and split them into 5 equal parts, calculating their average PD separately. There is an obvious trend that data with higher PD show greater consistency in GPT-4 annotations, and the consistency between GPT-4 annotations and the original dataset labels, which is acc, is also higher. The data with the lowest 0-20% difference received less than 40% of the 12:0 outcomes. Additionally, it accounted for nearly 40% of the total outcomes ranging from 6:6 to 9:3. The accuracy of these outcomes was lower than 50%. While the 20% data with the highest difference obtained over 50% of 12:0 outcomes with 83% accuracy. This means PD indeed serves the effect of estimating p, for data with a higher PD indeed show greater annotation consistency.

PD Improves Reward Model's Calibrated Accuracy Prediction

In this section, we investigate the impact of data differences on the reward model. According to the previous findings, a dataset with fewer data carries a higher average difference and lower difference variance. Thus reward models trained using data with high PD should have better calibrated accuracy. To prove this, we tested our reward models obtained from previous experiments. We compute the results on our HH-RLHF test set for each reward model, and calculate the BT model calibrated prediction of accuracy $\frac{1}{1+e^{-\Delta}}$ (Bai et al. 2022a), where Δ is the score difference $R_\phi(x, y_1) - R_\phi(x, y_2)$. As shown in Figure 4, these reward models achieve similar actual accuracy on the test set. But in terms of the calibrated accuracy, the reward models trained on filtering data based on difference achieve a higher calibrated accuracy as the overall data quantity drops, while the calibrated accuracy of the reward models trained on randomly sampled data decreases as the amount of data decreases. This confirmed our conclusion that low-difference data impairs the performance of the reward model.

Improving RLHF Performance by Filtering with PD

We experimented with the impact of difference on RL model performance. First, we trained default reward mod-

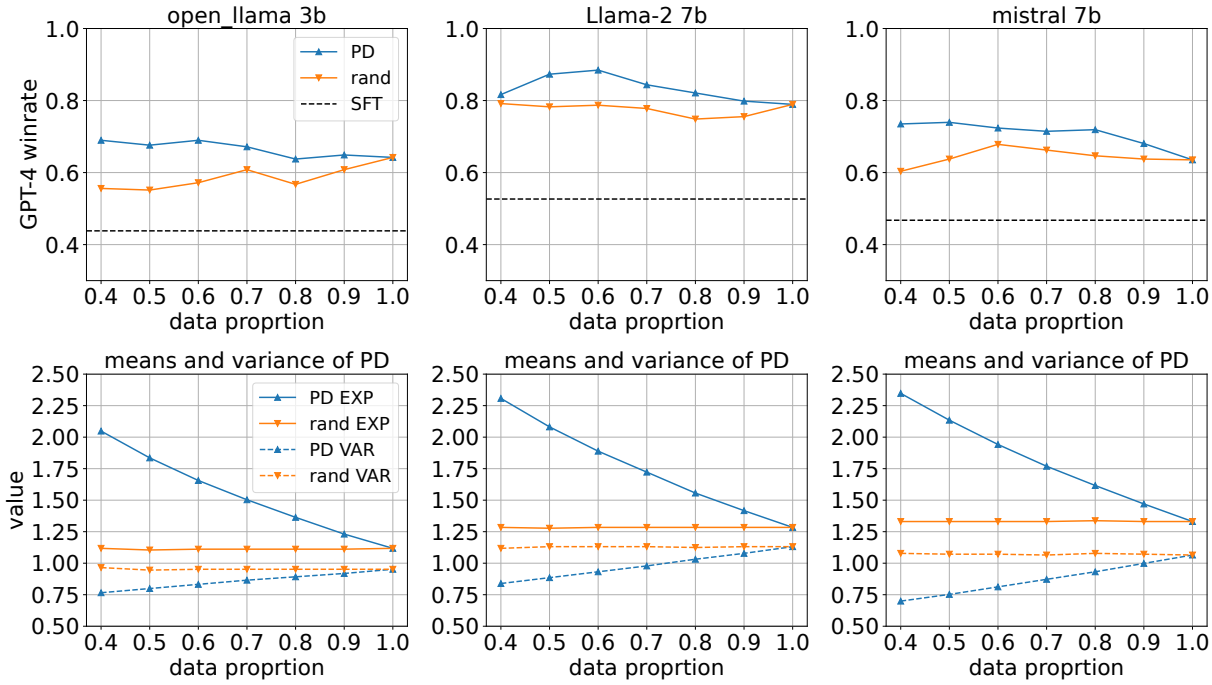

Figure 5: Data proportion vs. GPT-4 win rates for HH-RLHF test set. 512 samples were evaluated for data proportion and win rates. **Three columns** are results of Llama-2-7b, mistral-7b and open_Llama_3b_v2 models. **First row** is GPT-4 (gpt-4-1106-preview) winrates on overall test set. **PD** means filter data with difference, **rand** means random sampling and **sft** means the SFT model. **Second row** show the average difference **EXP** and difference variance **VAR** of each reward models' training set. The optimal results of above models under different settings are displayed in Table 1. More details are listed in the appendix.

els and RL models on the origin train set as our full data baseline. Then we computed the PD for all data used for reward model training. We sorted the pairs of samples based on their difference from high to low and then filtered the data from the front according to different ratios $w \in (0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0)$ [1]. We maintain the ratio of helpful and harmless data in the filtered data consistent with the ratio in the original data, to prevent the training data from being biased towards a particular dimension.

Under this setting, the dataset filtered by PD carries higher average PD and lower PD variance as shown in Figure 5. To serve as a control, we also used a randomly sampled dataset of the same size. Our main results are shown in Figure 5. The optimal results of each model under different settings are displayed in Table 1. In the experiment results of all three models, training the reward model with data that has a larger difference achieves better RLHF performances than those trained on equivalent random sampling data across nearly every percentage of filtering. In nearly all cases, these results even exceed the default reward model trained with full data ($w = 1.0$). We noticed that within the range of $w \in [0.6, 1.0]$, the overall performance shows an upward trend, which proved that the best trade-off ratio is between 50% and 60% on the HH-RLHF dataset. The performance drop in the range of $w \in [0.4, 0.6]$ is, we believe, due to the high consistency of preference in the data used for training the reward model, leading the reward model to learn a relatively simplified preference, thereby reducing overall RLHF performances. More details and extended results can be found in appendix.

From the perspective of harmfulness and helpfulness, the experimental results are more stable in terms of harmfulness, and the effect of data filtering according to the difference is also stronger. Besides, when w reaches above 0.5, there is no significant bias towards either helpfulness or harmlessness. Our method of filtering data based on the difference does not make the model more inclined towards each dimension.

PD is Effective for DPO

We use a DPO model trained on the HH-RLHF dataset on the Llama-7B SFT model to calculate $\mathbf{PD}_{\text{DPO}}$. For $\mathbf{PD}_{\text{DPO}}$ utilizing, we filter the data from high to low PD and include a random control. The results are shown in Table 2. The method of using PD filtering to select data achieves the highest win rate for GPT-4, which proves PD's effectiveness for DPO algorithm. More details are listed in the appendix.

PD's influence on Model's Latent Biases

As previous works (Ouyang et al. 2022; Bai et al. 2022a) highlight, it is pretty challenging to capture human preference due to the subjectivity and ambiguity of human judgments. If not carefully tuned, a reward model can be over-

[1] We attempted to filter the data from low to high PD, resulting in reward models with accuracy dropping below 50%

Model	Setting	winrate(%)			Proportion	Var(PD)	EXP(PD)
		helpful	harmless	overall			
open_llama_3B_v2	SFT	45.31	42.97	44.14	-	-	-
	full	52.34	77.34	64.84	100%	0.950	1.111
	random	50.78	72.27	61.52	70%	0.965	1.117
	PD	**58.98**	**80.86**	**69.92**	40%	0.768	2.046
Llama-2-7B	SFT	63.28	42.97	53.12	-	-	-
	full	74.21	85.94	80.08	100%	1.129	1.280
	random	71.88	89.06	80.47	40%	1.121	1.280
	PD	**86.32**	**93.36**	**89.84**	60%	0.930	1.886
Mistral-7B	SFT	50.78	43.36	47.07	-	-	-
	full	60.55	67.97	64.26	100%	1.070	1.332
	random	60.94	76.56	68.75	60%	1.068	1.330
	PD	**65.62**	**84.38**	**75.00**	50%	0.755	2.132

Table 1: The best results of different models for data filtering based on random and PD, selecting the optimal results from each different schemes of each model in Figure 5

Setting	winrate(%)			Proportion
	helpful	harmless	overall	
SFT	63.28	42.97	53.12	-
full	74.84	58.79	66.82	100%
random	69.53	54.88	62.21	90%
PD	**79.10**	**68.16**	**73.63**	60%

Table 2: The best results of Llama-2-7B DPO model, selecting the optimal results from each different scheme.

optimized (Gao, Schulman, and Hilton 2022) and potentially lead to risks of emergent deception (Perez et al. 2022). We followed Anthropic (Perez et al. 2022), using the Model-Written Evaluation Datasets (Perez et al. 2022) to conduct a risk assessment on the model filtered by PD, including Persona, Sycophancy, Advanced AI Risks. The overall results for different Llama-2 reward models are shown in Figure 6. We found that, with changes in data proportion, the fluctuation of risk bias was relatively significant. However, overall, the reward model filtered by PD showed lower preference for bias-matching behavior compared to the baseline. The PD filtering method improves alignment performance while essentially maintaining the original level of risk bias.

Conclusions

In this paper, we conducted an in-depth analysis of how the difference of preference data influences the reward model, and introduce PD metric to help filter the training data to get better RLHF performance. We start from the BT probability, viewing the annotation process of the preference data as a binomial distribution. We discovered the weight bias in empirical loss by deriving the expected loss. By computing the MSE between the expected loss and the empirical loss, we proved that data with higher and more consistent differences help reduce the bias. We introduce a definition for PD to help filter data due to the substantial cost of extensive annotation. We verified the reliability of PD by computing multiple GPT-4 judgment agreements. We conducted our experiments on HH-RLHF and 3 different models and got much better performances in reward model calibrated accuracy and the safety alignment task performance. Further,

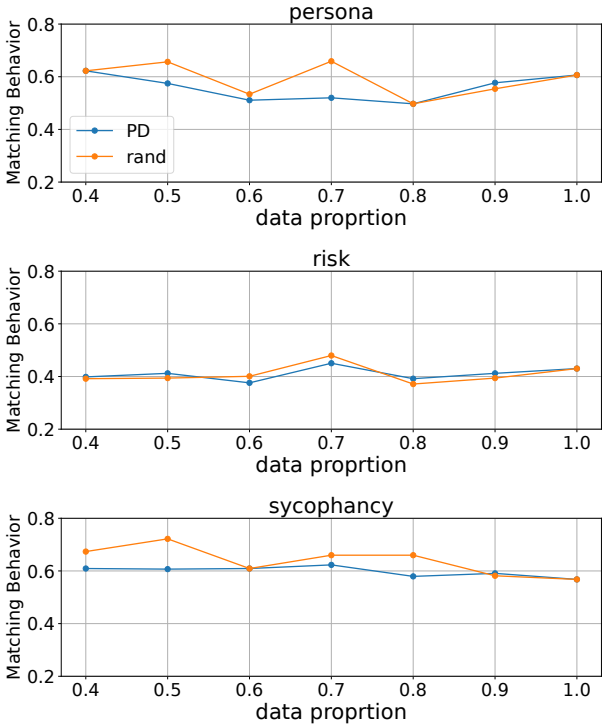

Figure 6: Results on persona, sycophancy, and advanced AI risk datasets for Llama-2-7B reward models. A higher value of matching behavior signifies a higher risk or bias.

we also verified the effectiveness of PD on DPO. Last we discussed if PD intensifies AI risk and bias. We proved that higher and more consistent data differences improve RLHF performance, achievable via PD filtering.

Acknowledgments

We would like to thank anonymous reviewers for their suggestions and comments sincerely. The work was partially supported by the Beijing Natural Science Foundation (L247010) and National Natural Science Foundation of

China (NSFC62076032).

References

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T. J.; Joseph, N.; Kadavath, S.; Kernion, J.; Conerly, T.; El-Showk, S.; Elhage, N.; Hatfield-Dodds, Z.; Hernandez, D.; Hume, T.; Johnston, S.; Kravec, S.; Lovitt, L.; Nanda, N.; Olsson, C.; Amodei, D.; Brown, T. B.; Clark, J.; McCandlish, S.; Olah, C.; Mann, B.; and Kaplan, J. 2022a. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. *ArXiv*, abs/2204.05862.

Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; Chen, C.; Olsson, C.; Olah, C.; Hernandez, D.; Drain, D.; Ganguli, D.; Li, D.; Tran-Johnson, E.; Perez, E.; Kerr, J.; Mueller, J.; Ladish, J.; Landau, J.; Ndousse, K.; Lukoiūtė, K.; Lovitt, L.; Sellitto, M.; Elhage, N.; Schiefer, N.; Mercado, N.; DasSarma, N.; Lasenby, R.; Larson, R.; Ringer, S.; Johnston, S.; Kravec, S.; Showk, S. E.; Fort, S.; Lanham, T.; Telleen-Lawton, T.; Conerly, T.; Henighan, T. J.; Hume, T.; Bowman, S.; Hatfield-Dodds, Z.; Mann, B.; Amodei, D.; Joseph, N.; McCandlish, S.; Brown, T. B.; and Kaplan, J. 2022b. Constitutional AI: Harmlessness from AI Feedback. *ArXiv*, abs/2212.08073.

Beeching, E.; Fourrier, C.; Habib, N.; Han, S.; Lambert, N.; Rajani, N.; Sanseviero, O.; Tunstall, L.; and Wolf, T. 2023. Open LLM Leaderboard. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard.

Bradley, R. A.; and Terry, M. E. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4): 324–345.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Cai, Z.; Cao, M.; Chen, H.; Chen, K.; Chen, K.; Chen, X.; Chen, X.; Chen, Z.; Chen, Z.; Chu, P.; Dong, X.; Duan, H.; Fan, Q.; Fei, Z.; Gao, Y.; Ge, J.; Gu, C.; Gu, Y.; Gui, T.; Guo, A.; Guo, Q.; He, C.; Hu, Y.; Huang, T.; Jiang, T.; Jiao, P.; Jin, Z.; Lei, Z.; Li, J.; Li, J.; Li, L.; Li, S.; Li, W.; Li, Y.; Liu, H.; Liu, J.; Hong, J.; Liu, K.; Liu, K.; Liu, X.; Lv, C.; Lv, H.; Lv, K.; Ma, L.; Ma, R.; Ma, Z.; Ning, W.; Ouyang, L.; Qiu, J.; Qu, Y.; Shang, F.; Shao, Y.; Song, D.; Song, Z.; Sui, Z.; Sun, P.; Sun, Y.; Tang, H.; Wang, B.; Wang, G.; Wang, J.; Wang, J.; Wang, R.; Wang, Y.; Wang, Z.; Wei, X.; Weng, Q.; Wu, F.; Xiong, Y.; Xu, C.; Xu, R.; Yan, H.; Yan, Y.; Yang, X.; Ye, H.; Ying, H.; Yu, J.; Yu, J.; Zang, Y.; Zhang, C.; Zhang, L.; Zhang, P.; Zhang, P.; Zhang, R.; Zhang, S.; Zhang, S.; Zhang, W.; Zhang, W.; Zhang, X.; Zhang, X.; Zhao, H.; Zhao, Q.; Zhao, X.; Zhou, F.; Zhou, Z.; Zhuo, J.; Zou, Y.; Qiu, X.; Qiao, Y.; and Lin, D. 2024. InternLM2 Technical Report. arXiv:2403.17297.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Cui, G.; Yuan, L.; Ding, N.; Yao, G.; He, B.; Zhu, W.; Ni, Y.; Xie, G.; Xie, R.; Lin, Y.; Liu, Z.; and Sun, M. 2024. UltraFeedback: Boosting Language Models with Scaled AI Feedback. arXiv:2310.01377.

Dai, J.; Pan, X.; Sun, R.; Ji, J.; Xu, X.; Liu, M.; Wang, Y.; and Yang, Y. 2023. Safe rlhf: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773*.

Dubois, Y.; Li, X.; Taori, R.; Zhang, T.; Gulrajani, I.; Ba, J.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2024. AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback. arXiv:2305.14387.

Gao, L.; Schulman, J.; and Hilton, J. 2022. Scaling Laws for Reward Model Overoptimization. arXiv:2210.10760.

Gao, L.; Schulman, J.; and Hilton, J. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, 10835–10866. PMLR.

Geng, X.; and Liu, H. 2023. OpenLLaMA: An Open Reproduction of LLaMA.

Ji, J.; Liu, M.; Dai, J.; Pan, X.; Zhang, C.; Bian, C.; Zhang, C.; Sun, R.; Wang, Y.; and Yang, Y. 2023. BeaverTails: Towards Improved Safety Alignment of LLM via a Human-Preference Dataset. arXiv:2307.04657.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; Lavaud, L. R.; Lachaux, M.-A.; Stock, P.; Scao, T. L.; Lavril, T.; Wang, T.; Lacroix, T.; and Sayed, W. E. 2023. Mistral 7B. arXiv:2310.06825.

OpenAI. 2023. GPT-4 Technical Report. *ArXiv*, abs/2303.08774.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744.

Perez, E.; Ringer, S.; Lukošiūtė, K.; Nguyen, K.; Chen, E.; Heiner, S.; Pettit, C.; Olsson, C.; Kundu, S.; Kadavath, S.; Jones, A.; Chen, A.; Mann, B.; Israel, B.; Seethor, B.; McKinnon, C.; Olah, C.; Yan, D.; Amodei, D.; Amodei, D.; Drain, D.; Li, D.; Tran-Johnson, E.; Khundadze, G.; Kernion, J.; Landis, J.; Kerr, J.; Mueller, J.; Hyun, J.; Landau, J.; Ndousse, K.; Goldberg, L.; Lovitt, L.; Lucas, M.; Sellitto, M.; Zhang, M.; Kingsland, N.; Elhage, N.; Joseph, N.; Mercado, N.; DasSarma, N.; Rausch, O.; Larson, R.; McCandlish, S.; Johnston, S.; Kravec, S.; El Showk, S.; Lanham, T.; Telleen-Lawton, T.; Brown, T.; Henighan, T.; Hume, T.; Bai, Y.; Hatfield-Dodds, Z.; Clark, J.; Bowman, S. R.; Askell, A.; Grosse, R.; Hernandez, D.; Ganguli, D.; Hubinger, E.; Schiefer, N.; and Kaplan, J. 2022. Discovering Language Model Behaviors with Model-Written Evaluations.

Rafailov, R.; Sharma, A.; Mitchell, E.; Ermon, S.; Manning, C. D.; and Finn, C. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. arXiv:2305.18290.

Rasley, J.; Rajbhandari, S.; Ruwase, O.; and He, Y. 2020. Deepspeed: System optimizations enable training deep

learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 3505–3506.

Sadigh, D.; Dragan, A. D.; Sastry, S.; and Seshia, S. A. 2017. *Active preference-based learning of reward functions*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. F. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33: 3008–3021.

Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; Bikel, D.; Blecher, L.; Ferrer, C. C.; Chen, M.; Cucurull, G.; Esiobu, D.; Fernandes, J.; Fu, J.; Fu, W.; Fuller, B.; Gao, C.; Goswami, V.; Goyal, N.; Hartshorn, A.; Hosseini, S.; Hou, R.; Inan, H.; Kardas, M.; Kerkez, V.; Khabsa, M.; Kloumann, I.; Korenev, A.; Koura, P. S.; Lachaux, M.-A.; Lavril, T.; Lee, J.; Liskovich, D.; Lu, Y.; Mao, Y.; Martinet, X.; Mihaylov, T.; Mishra, P.; Molybog, I.; Nie, Y.; Poulton, A.; Reizenstein, J.; Rungta, R.; Saladi, K.; Schelten, A.; Silva, R.; Smith, E. M.; Subramanian, R.; Tan, X. E.; Tang, B.; Taylor, R.; Williams, A.; Kuan, J. X.; Xu, P.; Yan, Z.; Zarov, I.; Zhang, Y.; Fan, A.; Kambadur, M.; Narang, S.; Rodriguez, A.; Stojnic, R.; Edunov, S.; and Scialom, T. 2023b. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

Wang, B.; Zheng, R.; Chen, L.; Liu, Y.; Dou, S.; Huang, C.; Shen, W.; Jin, S.; Zhou, E.; Shi, C.; et al. 2024. Secrets of rlhf in large language models part ii: Reward modeling. *arXiv preprint arXiv:2401.06080*.

Zheng, R.; Dou, S.; Gao, S.; Hua, Y.; Shen, W.; Wang, B.; Liu, Y.; Jin, S.; Liu, Q.; Zhou, Y.; Xiong, L.; Chen, L.; Xi, Z.; Xu, N.; Lai, W.; Zhu, M.; Chang, C.; Yin, Z.; Weng, R.; Cheng, W.; Huang, H.; Sun, T.; Yan, H.; Gui, T.; Zhang, Q.; Qiu, X.; and Huang, X. 2023. Secrets of RLHF in Large Language Models Part I: PPO. arXiv:2307.04964.

Ziegler, D. M.; Stiennon, N.; Wu, J.; Brown, T. B.; Radford, A.; Amodei, D.; Christiano, P.; and Irving, G. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.