On Diversified Preferences of Large Language Model Alignment

Anonymous ACL submission

Abstract

Aligning large language models (LLMs) with human preferences has been recognized as the key to improving LLMs' interaction quality. However, in this pluralistic world, human preferences can be diversified due to annotators' different tastes, which hinders the effectiveness of LLM alignment methods. This paper presents the first quantitative analysis of commonly used human feedback datasets to investigate the impact of diversified preferences on reward modeling. Our analysis reveals a correlation between the calibration performance of reward models (RMs) and the alignment performance of LLMs. We find that diversified preference data negatively affect the calibration performance of RMs on human-shared preferences, such as *Harmless&Helpful*, thereby impairing the alignment performance of LLMs. To address the ineffectiveness, we propose a novel Multi-Objective Reward learning method (MORE) to enhance the calibration performance of RMs on shared preferences. We validate our findings by experiments on three models and five human preference datasets. Our method significantly improves the prediction calibration of RMs, leading to better alignment of the Alpaca-7B model with *Harmless&Helpful* preferences. Furthermore, the connection between reward calibration and preference alignment performance suggests that calibration error can be adopted as a key metric for evaluating RMs.

1 Introduction

Large language models (LLMs), such as ChatGPT (OpenAI, 2023) and LLaMa (Touvron et al., 2023a,b), have significantly accelerated the development process toward artificial general intelligence (AGI). Among the key factors for such great achievement, the *alignment* technique, which finetunes LLMs with human feedback (Christiano et al., 2017), has played an essential role in training LLMs' responses to follow human values (*e.g.*, helpfulness and harmlessness) (Askell et al., 2021). Among the LLM alignment algorithms, reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022) has become the mainstream solution, which first learns a reward model (RM) representing human preferences and then updates LLMs via the proximal policy optimization (PPO) (Schulman et al., 2017) toward generating responses with higher RM scores. Alternative alignment methods also have been sequentially proposed for better computational complexity and training instability, such as RAFT (Dong et al., 2023b), DPO (Rafailov et al., 2023), RRHF (Yuan et al., 2023), and APO (Cheng et al., 2023b).

The performance of these alignment methods highly depends on the quality of human preference data $(\boldsymbol{x}, \boldsymbol{y}_w, \boldsymbol{y}_l)$, where $\boldsymbol{x}$ is the input query to the LLM, and response $\boldsymbol{y}_w$ is preferred to response $\boldsymbol{y}_l$ under the human annotation (Ouyang et al., 2022). Ideally, the preference datasets should *uniformly* be helpful, harmless, benevolent, and unbiased to guide the LLM alignment. However, in real-world scenarios, individuals can have *diversified* preferences on the same topic based on their different experiences, educational backgrounds, religions, and cultures (Leonardelli et al., 2021). Even for the same person, his or her expected model answer to a particular question can vary depending on different scenarios (Cheng et al., 2023a). The annotation disagreement, which is caused by different annotators or the same annotator in different scenarios (Bai et al., 2022), will significantly hinder the effectiveness of alignment methods (Davani et al., 2022; Wan et al., 2023).

To identify the diversified preferences quantitatively, we select five commonly used human feedback datasets, train an RM on each, and then test the performance on the other sets (details in Section 3). We plot the observation results in Figure 1. We observe that training RM on a single preference data source may cause inconsistent reward distribution shifts (middle plot), result in diverse

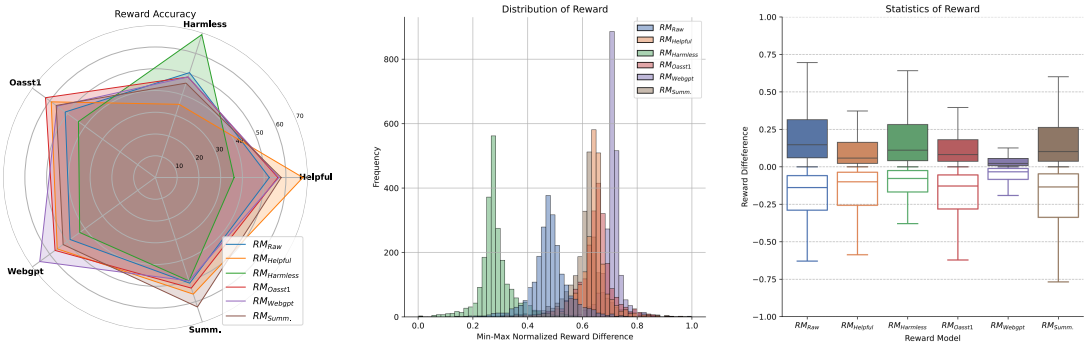


Figure 1: Illustration of *Diversified Preferences*. **Left:** reward accuracy on each preference. **Middle:** the reward distribution of each RM on harmless preference. **Right:** the reward statistics of each RM on harmless preference. The solid box indicates the reward statistics on correct rewarded samples, and the hollow box indicates the wrong rewarded samples.

reward values (right plot), and compromise the performance of other sets (left plot). The result indicates that different human preference datasets have different preference distributions (Cheng et al., 2023a). Hence, a more comprehensive understanding of the impact of diversified human preference datasets on the reward model becomes crucial, yet it has not received adequate attention and remained unexplored in the LLM alignment domain.

In our exploration, we found the *over-rewarding* phenomenon, that is, the vanilla RMs tend to output extreme rewards on samples, which damages the RMs and LLM alignment. To enhance the efficiency of leveraging the diversified preference datasets, inspired by multi-objective optimization methods (Sener and Koltun, 2018; Zeng et al., 2023), we regard RMs as a *shared reward* additionally with a customized *reward drift*. The shared reward represents the shared preferences across datasets (or general human preferences) and the reward drift contains individual or domain-specific preference information (Cheng et al., 2023a). Then, we introduce a **M**ulti-**O**bjective **Re**ward training scheme (MORE) to capture the shared (general) preference information, which adopts a novel reweight techniques to minimize the mean gradient of enlarging reward drifts. With MORE, RMs can capture a broader range of preferences and mitigate the impact of reward drifts. The main contributions of this paper are:

- This is the first work to demonstrate the positive correlation between the calibration performance of RMs and the alignment performance of LLMs. Moreover, RM learning on diversified preferences typically induces high calibration errors, indicating unreliable rewards. The unreliable rewards come from a *over-rewarding* phenomenon, denoting vanilla RMs output extreme rewards inducing harmful *reward drifts*. Hence, it negatively impacts the performance of LLM alignment.

- We induce a simple and effective Multi-Objective Reward (MORE) training scheme to alleviate the over-rewarding phenomenon. MORE makes self-adaption to the RM learning gradient to mitigate the *reward drifts*. MORE effectively enhances the calibration performance of RMs, especially on shared preferences across diversified preference datasets.

- We verified our findings with Pythia-1.4B, Pythia-2.8B (Biderman et al., 2023) and LLaMa2-7B (Touvron et al., 2023b) on **five** widely recognized and diverse preference datasets. Through empirical analysis, we established that MORE significantly minimizes reward drift and achieves low *Expected Calibration Error* (ECE) values. Additionally, by applying reject sampling to Alpaca-7B (Taori et al., 2023) with the RMs generated, we aligned the models with *Helpful&Harmless* preferences, thereby affirming the critical role of ECE in the evaluation of Reward Models.

2 Background

Large language Model Alignment Parameterized by $\boldsymbol{\theta}$, a reward model (RM) is a mapping $r_{\boldsymbol{\theta}} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, which provides a real-valued reward score $r_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y})$ evaluating a textual response $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_M) \in \mathcal{Y}$ corresponding to an input prompt $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N) \in \mathcal{X}$. Given a sample $(\boldsymbol{x}, \boldsymbol{y}_w, \boldsymbol{y}_l) \sim \mathcal{D}$ from a preference dataset $\mathcal{D}$, $r_{\boldsymbol{\theta}}$ is expected to provide a preference score with $r_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}_w) > r_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}_l)$, representing the response $\boldsymbol{y}_w$ is preferred. Following the Bradley-Terry model (David, 1963), the RM learning objective on the preference dataset $(\boldsymbol{x}, \boldsymbol{y}_w, \boldsymbol{y}_l) \sim \mathcal{D}$ is

defined as:

$$\mathcal{L}_{\text{rank}}(\boldsymbol{\theta}; \mathcal{D}) = -\mathbb{E}_{\mathcal{D}}\left[\log(\sigma\left(\Delta r_{\boldsymbol{\theta}}(\boldsymbol{y}_w, \boldsymbol{y}_l)\right))\right] \quad (1)$$

where we use $\Delta r_{\boldsymbol{\theta}}(\boldsymbol{y}_w, \boldsymbol{y}_l)$ to denote reward difference $r_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}_w) - r_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}_l)$ for simplifying notation in this paper and $\sigma(\cdot)$ is the Sigmoid function. With a well-learned reward $r_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y})$, LLM alignment optimizes the generation policy $\pi(\boldsymbol{y}|\boldsymbol{x})$ by maximizing the expected reward value:

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}, \boldsymbol{y}\sim\pi(\boldsymbol{y}|\boldsymbol{x})}[r_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y})] \\ &\quad - \beta\mathbb{D}_{\text{KL}}[\pi(\boldsymbol{y}|\boldsymbol{x})\|\pi_{\text{ref}}(\boldsymbol{y}|\boldsymbol{x})],\end{aligned} \quad (2)$$

where $\mathbb{D}_{\text{KL}}[\pi(\boldsymbol{y}|\boldsymbol{x})\|\pi_{\text{ref}}(\boldsymbol{y}|\boldsymbol{x})]$ is the KL divergence regularizer between current policy π and a reference π_{ref}, preventing the optimization from instability and degeneration. The typical solution to the preference optimization in equation 3 is reinforcement learning (RLHF) (Ouyang et al., 2022), especially with the proximal policy optimization (PPO) algorithms (Schulman et al., 2017). However, RLHF has been recognized as practically suffering from implementation complexity and training instability. To avoid the RL schedule during alignment, reject sampling methods (Liu et al., 2023) directly conduct supervised fine-tuning on $\boldsymbol{y}^{\text{best}}$ to further simplify the human preference alignment process. The rejection sampling optimization (RJS) loss can be written as

$$\mathcal{L}_{\text{RJS}}(\pi) = -\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}, \boldsymbol{y}\sim\pi(\boldsymbol{y}|\boldsymbol{x})}[\log \pi(\boldsymbol{y}^{\text{best}}|\boldsymbol{x})], \quad (3)$$

where $\boldsymbol{y}^{\text{best}} = \arg\max_{1\le s\le S\{r(\boldsymbol{x}, \boldsymbol{y}^s)\}}$ is the sampled response with the highest reward score.

Calibration Error Calibration error is an effective method to estimate the confidence of a model's outputs (Guo et al., 2017). Numerous studies have focused on improving the calibration performance of statistical machine-learning systems (DeGroot and Fienberg, 1983; Palmer et al., 2008; Yang and Thompson, 2010). Furthermore, the calibration error of neural networks provides additional information for users to determine whether to trust the model's predictions, especially for modern neural networks that are more challenging to interpret (Guo et al., 2017; Zhu et al., 2023). In the field of natural language processing, studies have revealed a positive relationship between calibration performance and the reduction of hallucination (Xiao and Wang, 2021; Tian et al., 2019), and the evaluation of pre-trained language models (Kadavath et al., 2022; Tian et al., 2023). The calibration error has demonstrated its ability to evaluate the performance of language models. We provide its computation in the Appendix A. In this paper, we first employ the calibration error to evaluate the RMs. Subsequently, we investigate the implicit connection between RMs and LLM alignment under diversified preferences.

3 Empirical Study of Diversified Preferences

We start with an empirical analysis of diversified preferences in reward modeling on multiple sources $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_K\}$, where each data source D_k contains the preference comparison pairs from different tasks (Dong et al., 2023a), domains (Cheng et al., 2023a), or individuals (Bai et al., 2022). In this paper, we selected *Summarize* (Stiennon et al., 2020), *Webgpt* (Nakano et al., 2021a), *Helpful&Harmless* (Bai et al., 2022), and *OASST1* (Köpf et al., 2023) as the different preference sources to empirical analysis the phenomena of diversified preferences. We use Pythia-1.4B (Biderman et al., 2023) as the RM base, and finetuned RMs with comparisons from each source. The experiment setup aligns with Section 5.

The reward distributions across various RMs exhibit diversity when applied to the same dataset. We analyze and present the variation in rewards (defined as the difference in reward values assigned by an RM to the winning and losing samples) offered by these RMs, as illustrated in Figure 1 (additional results in Figure 8 and 9 in Appendix). Compared with the results of raw model RM_{Raw}, we observe that training on different datasets results in diverse reward values (right plot) and distribution shift (middle plot). Specifically, the reward value distribution of $\text{RM}_{\text{Harmless}}$ shifts from the RM_{Raw} in a certain degree. While the reward value distributions of $\text{RM}_{\text{Helpful}}$, $\text{RM}_{\text{Webgpt}}$, $\text{RM}_{\text{Oasst1}}$ and $\text{RM}_{\text{Summ.}}$ shifts to the a different direction. Moreover, despite the distribution of $\text{RM}_{\text{Helpful}}$, $\text{RM}_{\text{Webgpt}}$, $\text{RM}_{\text{Oasst1}}$ and $\text{RM}_{\text{Summ.}}$ are similar, the mean-variance of their reward values are quite different.

Furthermore, when considering the accuracy gains illustrated in Figure 1 (left plot), the observed shift in reward distribution indicates that the learned reward values from preference datasets are diversified. To effectively capture the shared reward values across these diversified preferences, it becomes necessary to formulate a new problem approach for reward modeling on diverse preference datasets.

4 Multi-Objective Reward Learning

In this section, we propose our reward modeling on diversified preference datasets, highlighting the implicit reward drift during the reward learning process and its negative impacts. Then, we present the MORE training schemes to mitigate the reward drifts as a feasible solution. To maintain the integrity of our paper, we leave our quantitative analyses of reward modeling on diversified preferences in the next section.

4.1 Preference Diversity as Reward Drift

We denote $r^*(\cdot,\cdot)$ as the shared reward function, which (ideally) provides reward values reflecting the general values among people (or shared preference information across datasets in practice). As the collected human-feedback datasets are limited and implicitly biased, training an RM $r_{\boldsymbol{\theta}}$ on a limited preference dataset can be viewed as drifting from an optimal reward. We can form a reward model $r_{\boldsymbol{\theta}}(\cdot,\cdot)$ with reward drift in a data level:

$$r_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}) = r^*_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}) + \tilde{r}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}), \quad (4)$$

where $\boldsymbol{x},\boldsymbol{y} \in \mathcal{X}\times\mathcal{Y}$, and $\tilde{r}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y})$ is the reward drift learned by RM $r_{\boldsymbol{\theta}}(\cdot,\cdot)$. Then, we investigate the vanilla ranking loss for reward modeling. Substituting reward function in (1) with the drifted form (4), we have $\mathcal{L}_{\text{rank}}(\boldsymbol{\theta};\mathcal{D}) =$

$$-\mathbb{E}_{\mathcal{D}}[\log(\sigma(\Delta r^*_{\boldsymbol{\theta}}(\boldsymbol{y}_w,\boldsymbol{y}_l) + \Delta\tilde{r}_{\boldsymbol{\theta}}(\boldsymbol{y}_w,\boldsymbol{y}_l)))]. \quad (5)$$

Hence, updating the RM to minimize the rank loss will enlarge the reward differences (input of the Sigmoid function). Simultaneously, the reward drift is also enlarged, causing over-rewarding.

4.2 Reward Modeling on Diversified Data

Letting $\boldsymbol{\theta}$ be the RM trained on mixed diverse datasets $\mathcal{D} = \{\mathcal{D}_1,\ldots,\mathcal{D}_K\}$, the $r_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y})$ can be viewed as a multi-task learner with shared parameters (Sener and Koltun, 2018). Then, the reward value provided by $r_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y})$ can be decomposed into voting format weighted by an implicit λ:

$$r_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}) = r^*_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}) + \textstyle\sum_{i=1}^K \lambda_i \tilde{r}_{\boldsymbol{\theta}_i}(\boldsymbol{x},\boldsymbol{y}), \quad (6)$$

where the shared reward $r^*_{\boldsymbol{\theta}}(\cdot,\cdot)$ is the same with arbitrary λ, and $\tilde{r}_{\boldsymbol{\theta}_i}(\cdot,\cdot)$ is the reward drift. We interpret that the $\tilde{r}_{\boldsymbol{\theta}_i}(\cdot,\cdot)$ is provided by subset of parameters $\boldsymbol{\theta}_i$, representing the preferences from the i-th dataset $\mathcal{D}_i$. This reward value decomposition naturally holds in the model output level, despite the non-linear nature of neural networks.

Moreover, our formulation aligns with multi-task learning (Crawshaw, 2020) and multi-objective learning (Guardieiro, 2023) problems. For example, the $\boldsymbol{\theta}$ can be implemented as an ensemble model, where $\{\boldsymbol{\theta}_i\}, i\in[N]$ is the base models. Therefore, it is natural to adjust the weight λ in an ensemble manner (Coste et al., 2023; Jang et al., 2023; Touvron et al., 2023a; Eisenstein et al., 2023) to mitigate the reward drift such that $\min\sum_{i=1}^K \lambda_i\tilde{r}_{\boldsymbol{\theta}_i}(\boldsymbol{x},\boldsymbol{y})$. Compared with average rewards from multiple RMs (Jang et al., 2023; Eisenstein et al., 2023), we focus on training a single RM that learns the shared preference. We propose to reduce the model update on reward drift during RM training via linear scalarization (Barrett and Narayanan, 2008). Moreover, we provide further discussion on related manners in Section 7.

4.3 Training Scheme: MORE

MORE loss function Our analyses suggest finding proper weights λ for mitigating reward drifts. Then, we propose training RMs to capture the shared preference across multiple datasets with the following objective:

$$\mathcal{L}_{\text{MORE}}(\boldsymbol{\theta};\mathcal{D}) = \textstyle\sum_{i=1}^K \lambda_i \mathcal{L}_{\text{rank}}(\boldsymbol{\theta},\mathcal{D}_i), \quad (7)$$

where $\sum_i^K \lambda_i = 1, \lambda_i \geq 0$. Compared with vanilla ranking loss in (1), the above loss additionally focuses on the combination relation across preferences. The linear combination of loss functions is commonly adopted in deep learning methods to balance the interaction of different modules (Zhang et al., 2023; Kurin et al., 2022). Analogously, we treat each preference as an individual module and balance them wisely. Moreover, this formulation also covers several typical training cases. For example, directly mixing diverse preference datasets $\mathcal{D} = \{\mathcal{D}_1,\ldots,\mathcal{D}_K\}$ and training a RM implicitly induces $\lambda_i = |\mathcal{D}_i|/|\mathcal{D}|$ (McMahan et al., 2017; Ramé et al., 2024). Therefore, if the number of data samples from a single preference is greatly larger than other preferences, the RM is likely to drift to the preference with more samples. Excluding data quantity, the weight is also decided by the quality of data samples in the training process (Katharopoulos and Fleuret, 2018; Zhou and Wu, 2023). Neural network training typically provides a larger gradient for harder samples (Katharopoulos and Fleuret, 2018), therefore, leaning the RMs preferences drift to these hard samples. In practice, the quantity and quality variance in diversified datasets may require more hyper-parameter searching (Guo et al., 2024)

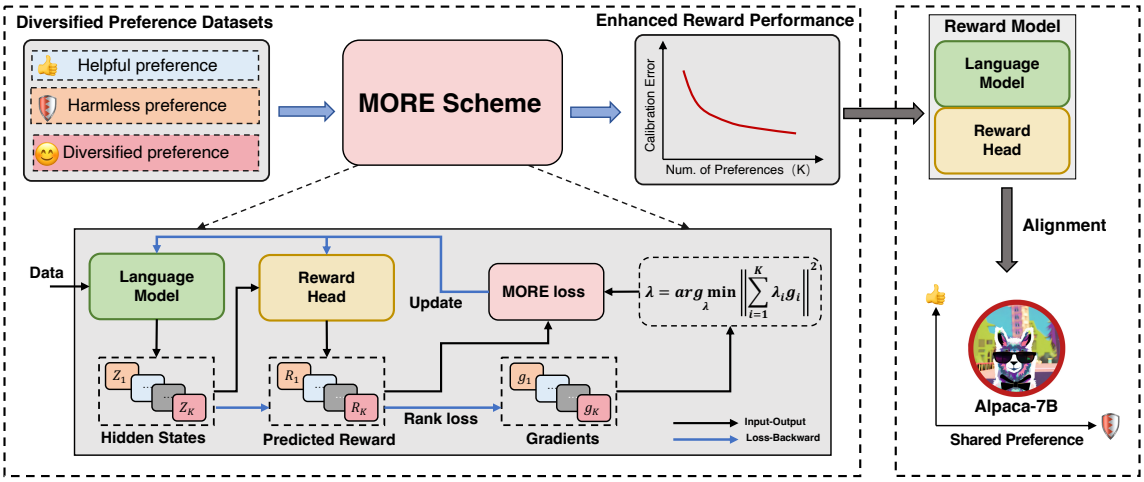


Figure 2: Multi-objective reward model training scheme (MORE), which consists of four steps: (1) collect a diversified batch of data from the mixed dataset; (2) calculate the RM gradient for each preference source; (3) minimize the reward drift to determine the scalar $(\lambda_1, \lambda_2, \ldots, \lambda_K)$ for MORE loss; (4) update the RM with the re-weighted RM loss. Lower calibration error indicates the RM provides an accurate reward.

or data composition efforts (Dong et al., 2023a) in the vanilla finetuning process.

What is MORE doing? We suggest training a better RM via self-adaption training weights λ for better data efficiency. The MORE loss minimizes the ranking loss by solving a reward drift mitigation task, applying a *batch-wise reweighting* method. Let batch data $\mathcal{B} = \{\boldsymbol{x}^{(b)}, \boldsymbol{y}_w^{(b)}, \boldsymbol{y}_l^{(b)}\}_{b=1}^{B} \sim \mathcal{D}$ be the sampled batch data from diverse datasets. Furthermore, $\mathcal{B}_i \sim \mathcal{D}_i \subset \mathcal{B}, \forall i \in [K]$ is the subset of batch data from the i-th preference dataset. We have the gradient $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{MORE}}(\boldsymbol{\theta}; \mathcal{B})$

$$= \sum_{b=1}^{B} \left[-\nabla_{\boldsymbol{\theta}} \log(\sigma(\Delta r^*_{\boldsymbol{\theta}}(\boldsymbol{y}_w^{(b)}, \boldsymbol{y}_l^{(b)}))) \right] + K \cdot \underbrace{\min \sum_{i=1}^{K} \lambda_i \sum_{j=1}^{|\mathcal{B}_i|} \left[-\nabla_{\boldsymbol{\theta}} \log(\sigma(\Delta \tilde{r}_{\boldsymbol{\theta}}(\boldsymbol{y}_w^{(j)}, \boldsymbol{y}_l^{(j)})) \right]}_{\text{Reward Drift Mitigation}}, \quad (8)$$

where we adjust λ to minimize the partial gradient of enlarging reward drifts. The mitigation task in (8) can be efficiently solved by the Frank-Wolfe solver (Jaggi, 2013; Sener and Koltun, 2018; Zhou et al., 2022b; Zeng et al., 2023). We provide the details of our efficient implementation in the Appendix B. Furthermore, $\mathcal{L}_{\text{MORE}}$ shares the same magnitude of vanilla loss function $\mathcal{L}_{\text{rank}}$ in expectation over the whole training dataset, as justified in Appendix B.

Outline The MORE only requires simple modification on batch data sampling and batch-wise reweighting. We depict the pipeline in Figure 2. MORE consists of THREE main steps as: 1) Sample a diverse batch data $\mathcal{B} = \{\mathcal{B}_i\}_{i=1}^{K}$, $\mathcal{B}_i = \{\boldsymbol{x}, \boldsymbol{y}_w, \boldsymbol{y}_l\}_{b=1}^{|\mathcal{B}_i|}$ and input the batch data forward the RM and obtain the hidden states $\{\boldsymbol{z}_i\}_{i=1}^{K}$, which is the inputs of the reward head $\boldsymbol{\theta}_{\text{rm}}$. 2) Compute the gradient of reward head with data $\{\boldsymbol{z}_i, \boldsymbol{y}_w, \boldsymbol{y}_l\}$. 3) Compute the weights λ by Frank-Wolfe solver. Finally, we substitute the loss weights in (7) as the final loss for the optimizer to conduct backward and model updating. This procedure prevents the RM from enlarging implicit reward drifts.

5 Experiments on Reward Modeling

In this section, we present our experiments and quantitative analyses on reward modeling.

Datasets & models We use open-sourced human preference alignment datasets, including *Helpful&Harmless* (Bai et al., 2022), *OASST1* (Köpf et al., 2023), *Webgpt* (Nakano et al., 2021a), and *Summarize* (Stiennon et al., 2020). We provide the statistics of the datasets and data composition in Appendix 3. Despite these datasets being released to human preference alignment, our study highlights the preference diversity across the datasets and its impacts on training RMs. We train Pythia-1.4B, Pythia-2.8B (Biderman et al., 2023) and LLaMa2-7B (Touvron et al., 2023b) as the LM base for RM training. We use the last token embedding of the output hidden states as the pooled hidden representation, then add one linear layer (RM head) with the scale-value output on it to predict reward scores. We present the details of the training setup in Appendix C.

Baselines We compare our method with conventional fine-tuning strategies for training language

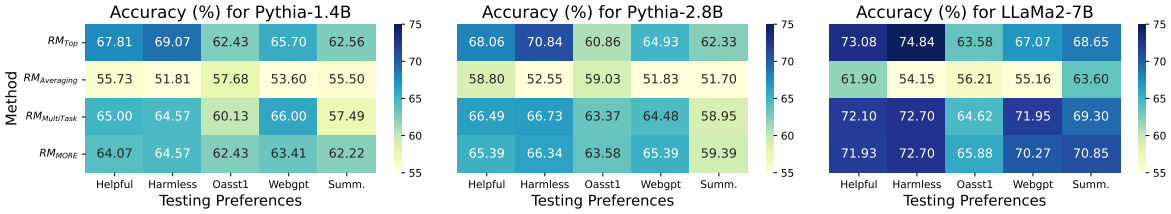


Figure 3: The reward accuracy of RMs with different training schemes on each dataset.

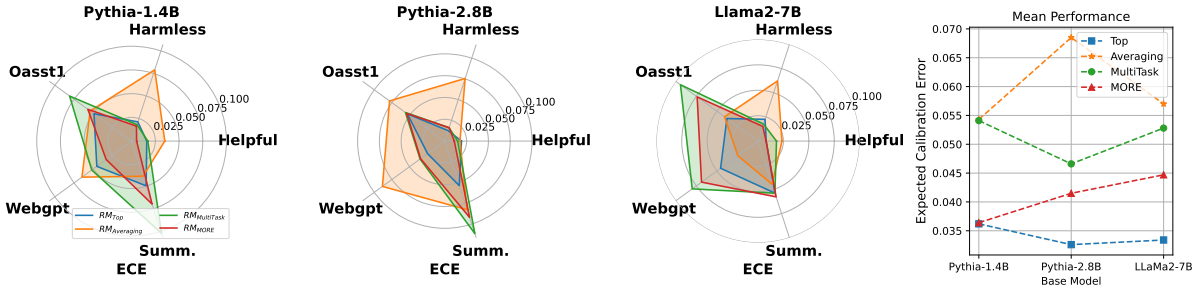


Figure 4: The ECE of the corresponding RMs.

models, specifically mixing the preference data samples. We refer to the training scheme as MultiTask training (Dong et al., 2023a). The **MultiTask** training scheme randomly samples data from hybrid preference datasets. Additionally, we compare with the **Top** performance of RMs trained on each preference dataset. We highlight that the **Top** performance indicates the ideal ensemble-RM, i.e., each sample obtains its reward from the corresponding best RM. Then, we naively **Average** the reward values from **Top** RMs provide on the same samples to denote a naive ensemble-RM. In all, we mark the baseline rewards as $\text{RM}_{\text{MultiTask}}$, RM_{Top} and $\text{RM}_{\text{Averaging}}$ respectively.

Evaluation metric We use the *preference accuracy* on test datasets for each domain. If an RM outputs $r(\boldsymbol{x}, \boldsymbol{y}_w) > r(\boldsymbol{x}, \boldsymbol{y}_l)$ for a test sample $(\boldsymbol{x}, \boldsymbol{y}_w, \boldsymbol{y}_l)$, we denote it as a correct prediction. The preference accuracy is then computed as the proportion of correct predictions within all testing response pairs. However, preference accuracy only provides pairwise comparisons of responses and does not reflect the degree of preference for each response. Following Bai et al. (2022); Cheng et al. (2023b), we examine the probability calibration to test if the learned RMs accurately represent the human preference distribution. This is measured by the *Expected Calibration Error* (Naeini et al., 2015; Zhu et al., 2023).

5.1 Reward Modeling on Diversified Preference Datasets

We provide the reward modeling results on mixed diversified datasets in Figure 3 and Figure 4. The detailed information is in Table 2 of the Appendix.

The reward accuracy does not drop significantly on mixed diversified preferences. Increasing the size of LLMs, reward model training on mixed diversified preference datasets can maintain reward accuracy. For instance, when Pythia-1.4B is used as the RM base model, the reward accuracy is lower compared to the Top accuracy achieved through single preference training on all preferences. Then, when LLaMa2-7B is used as the base model, the reward accuracy on the *Oasst1*, *Webgpt*, and *Summarise* test sets surpasses the top accuracy achieved through single training. Additionally, the degradation of reward accuracy on the *Helpful* and *Harmless* datasets is mitigated. Therefore, the performance of RMs typically is proportional to the size of base models (Gao et al., 2023). Moreover, we find the accuracy of $\text{RM}_{\text{Averaging}}$ is poor, revealing the preference conflicts across RM_{Top}.

Reward modeling on mixed diversified preferences affects calibration performance Noting the reward accuracy only provides comparisons of responses (Zhu et al., 2023), we emphasize the ECE performance reflects the degree of preference for responses in Figure 4. Compared $\text{RM}_{\text{MultiTask}}$ with RM_{Top}, reward modeling on mixing the diversified preference datasets typically degenerates calibration performance on all preferences. Especially, the reward accuracy of $\text{RM}_{\text{MultiTask}}$ and RM_{Top} are comparable but the calibration performances are very different. The LLMs can maintain high accuracy on all preferences due to their large capacity, however, the reward distribution is affected by mixed diversified preferences. *These*

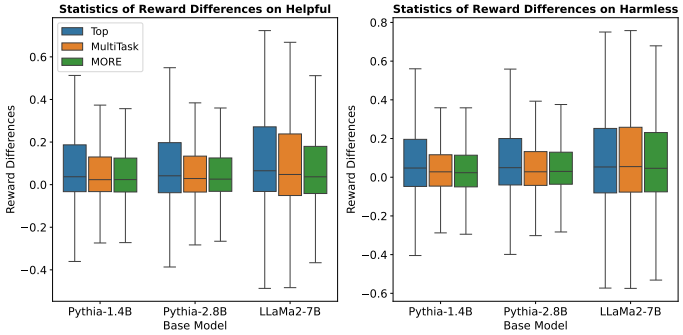


Figure 5: Reward differences on test samples. Positive reward differences indicate correct reward samples and negative reward differences indicate incorrect reward samples.

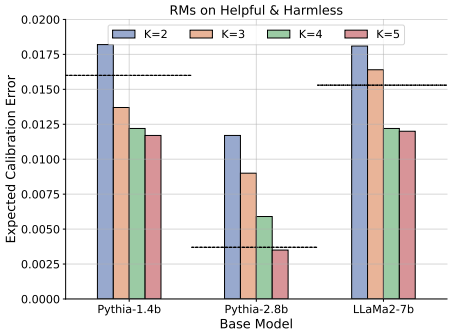


Figure 6: MORE enhance calibration performance with diversified preferences. The black dashes indicate the ECE of RM_{Top}.

findings reveal that reward accuracy is insufficient to verify the ability of RMs and suggest evaluation of RMs via ECE. We will further justify the point in the alignment experiments.

MORE implements significant calibration performance improvement The RM_{MORE} preserves a significantly lower ECE than $\text{RM}_{\text{MultiTask}}$, indicating that RM_{MORE} provides more accurate reward values. Moreover, RM_{MORE} implements significantly lower ECE than RM_{Top} on *Helpful&Harmless* preferences. This is because *Helpful&Harmless* preference is shared by these datasets and MORE accurately captures shared preferences across them. Therefore, MORE implements lower calibration errors on shared *Helpful&Harmless* preference and slightly loses its calibration performance on the other three preference datasets. This calibration performance gap between RM_{Top} and RM_{MORE} on the other three diversified preferences further reflects the preference diversity.

5.2 Analyses on RMs of *H&H* Preferences

To clarify the improvement of MORE, we provide analyses on *Helpful&Harmless* (*H&H*) datasets, which is an important human preference alignment objective for LLMs in recent works (Ouyang et al., 2022; Touvron et al., 2023b). Concretely, we focus on the statistics of the **reward difference** (i.e., $\Delta r_{\boldsymbol{\theta}}(\boldsymbol{y}_w, \boldsymbol{y}_l)$). We count the reward differences of RMs on *H&H* test datasets in Figure 5.

MORE mitigates *over-rewarding* phenomenon In Figure 5, we observe the RM_{Top} outputs large absolute reward differences on testing samples. On the contrary, the RM_{MORE} provides lower absolute reward differences on testing samples, compared with baseline training schemes. Moreover, RMs tend to provide extreme rewards to some samples. We count these extreme reward values as *outliners* in Appendix, Table 6. This phenomenon aligns with our methodology in (6), that is, MORE mitigated the reward drifting during training. Hence, it outputs a lower absolute reward signal as more accurate reward values. These findings reveal the phenomenon of *over-rewarding* in RMs, where vanilla RMs tend to assign large reward values to samples. This phenomenon demonstrates problem modeling (5). Importantly, the over-rewarding in RM may not break the reward accuracy shown in Figure 3, however, it induces unsatisfied calibration performance. MORE maintains the reward accuracy of RMs, alleviates the over-rewarding effects on reward modeling, and trains better RMs.

MORE achieves better calibration using more diversified preferences The MORE can benefit from diversified preference information by (8), which suggests increasing the number of diversified preferences can better mitigate reward drifts. We change the number of mixed preference datasets from 2 to 5 to verify our insights, as shown in Figure 6. In detail, we start from mixed *Helpful&Harmless* datasets (K=2) and then add *Oasst1, Webgpt, Summarise* datasets. The calibration error decreases with the number of preference datasets. It proves that MORE can utilize the preferences information to enhance the performance of the reward model on shared preferences and surprisingly outperforms RM_{Top}.

6 Experiments on LLM Alignment

In this section, we use the previously obtained RMs for LLM alignment experiments on Alpaca (Taori et al., 2023), which is an instruction-tuned LLaMA-7B model (Touvron et al., 2023a). We use Reject Sampling (RJS) (Touvron et al., 2023b; Liu et al., 2023) as the alignment algorithms, where we sample 4 responses from Alpaca with queries from *H&H* trainsets. Our experiment mainly justifies *the correlation between the calibration performance*

Reward Model				Perplexity (PPL)		GPT4 Evaluation (%)		
Base Model	Scheme	Acc(%)	ECE ↓	Helpful ↓	Harmless ↓	Win	Tie	Lose
-	-	-	-	15.48	12.71	-	-	-
Pythia-1.4B	MultiTask	64.79	0.0177	15.30	8.22	44	22	34
	MORE	64.32	0.0109	12.68	8.42			
Pythia-2.8B	MultiTask	66.61	0.0145	16.76	8.42	45	21	34
	MORE	65.87	0.0078	13.14	10.29			
LLaMa2-7B	MultiTask	72.40	0.0284	16.93	8.69	45	23	32
	MORE	72.32	0.0143	11.97	9.96			

Table 1: The RJS alignment performance with different RMs. The first line is the performance of Alpaca base model. The results show that ECE further reflects the ability of RMs when the reward accuracy is close.

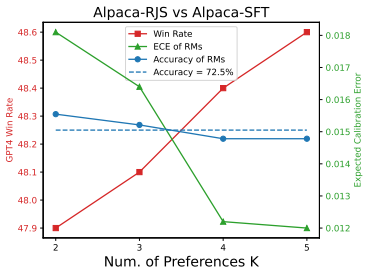


Figure 7: The correlation between ECE of RMs and RJS alignment performance for the Alpaca model.

of RMs and LLM alignment performance.

RM_{MORE} works better than $\text{RM}_{\text{MultiTask}}$ for RJS aligning *H&H* with lower ECEs We finetune Alpaca with the most preferred samples scored by previously obtained RMs to align the human preference of *H&H*, following RJS loss (3). We show the alignment performance in Table 1, where we use the same GPT4 evaluation prompts with DPO (Zhou et al., 2023) shown in Appendix D. RM_{MORE} works better for RJS tasks. Noting that RM_{MORE} and $\text{RM}_{\text{MultiTask}}$ implements comparable reward accuracy on *H&H*, while the calibration performance are significantly different. Therefore, *the alignment performance is additionally related to the calibration performance of the RMs.*

ECE of RMs is positively correlated with alignment performance We finetune the Alpaca model on the good response from *H&H* training datasets, and the finetuned model is marked by Alpaca-SFT. Then, we conduct the RJS alignment experiments with LLaMa2-7B RMs from Figure 6. We compare each alignment result of Alpaca-RJS models with the same Alpaca-STF model via GPT evaluation, shown in Figure 7. The results show that *the RMs with lower ECE values work better for RJS alignments, emphasizing the importance of calibration evaluation*.

7 Additional Discussions

Connections with data composition and ensemble-RM studies Dong et al. (2023a) have empirically shown that the LLM ability can be improved by adjusting the mixed training data ratio from different sources. However, the mixed proportion can be hard to search in practice. Besides, other studies have shown that direct ensemble RMs (Eisenstein et al., 2023) or merging RMs' parameters (Jang et al., 2023; Ramé et al., 2024) during training could also improve the ability of RMs. In practice, these approaches induce a large system burden for storing/training multiple RMs, especially since the RMs can be extremely large. In comparison, this paper focuses on training single RM on diversified datasets.

Suggestions for reward model training This paper reveals two main suggestions for future reward model training works. First, *Evaluate RMs with reward accuracy and calibration error.* Reward accuracy is insufficient to evaluate the ability of RMs due to model capacity and data quality. Our work suggests the community additionally focuses on the calibration performance of RMs. Besides, *Increasing the diversity of preference data samples can ensure the robustness of the reward modeling process.* Due to the preference information being typically noisy, learning reward information from mixed diversified datasets can be beneficial.

Applications The MORE can enhance preference modeling pre-trained (PMP) paradigm (Askell et al., 2021) as it captures the shared preference information. This facilitates its use in federated learning scenarios (McMahan et al., 2017), where the data distributions are highly heterogeneous across participants. Moreover, the RM_{MORE} can be easily finetuned to specific preferences (Cheng et al., 2023a). This flexibility allows for the adaptation of our approach to various applications.

Extension to RM-free alignment methods RM-free alignment methods (Rafailov et al., 2023; Azar et al., 2023) are derived based on an implicit reward model. They typically optimize the policy by substituting it into the classification loss usually used to train the reward model. *The relation of calibration performance of implicit reward and the alignment performance in the RM-free methods is unexplored.* Besides, learning shared preferences from mixed diverse preference datasets can be extended to RM-free paradigms. For example, we can re-weight the partial reward loss of the RM-free alignment methods, especially DPO (Rafailov et al., 2023; Zhou et al., 2023). We will explore this in future work.

8 Limitations

We only conducted experiments using the conventional RJS algorithm in LLM alignment tasks. As a reward modeling algorithm that captures shared preference information, MORE depends on the quality of the applied data. Therefore, the correlation of ECE of RMs and LLM alignment performance in other alignment algorithms requires further exploration. Besides, the training datasets we used contain violence, abuse, and biased content that can be upsetting or offensive to particular groups of people.

References

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2023. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Leon Barrett and Srini Narayanan. 2008. Learning all optimal policies with multiple criteria. In *Proceedings of the 25th international conference on Machine learning*, pages 41–47.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

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

Pengyu Cheng, Yifan Yang, Jian Li, Yong Dai, and Nan Du. 2023b. Adversarial preference optimization. *arXiv preprint arXiv:2311.08045*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

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

Michael Crawshaw. 2020. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*.

Gabriela Csurka. 2017. A comprehensive survey on domain adaptation for visual applications. *Domain adaptation in computer vision applications*, pages 1–35.

Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2022. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110.

Herbert Aron David. 1963. *The method of paired comparisons*, volume 12. London.

Morris H DeGroot and Stephen E Fienberg. 1983. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22.

Yuntian Deng, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc'Aurelio Ranzato. 2020. Residual energy-based models for text generation. *arXiv preprint arXiv:2004.11714*.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2023a. How abilities in large language models are affected by supervised fine-tuning data composition. *arXiv preprint arXiv:2310.05492*.

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

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, et al. 2023. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint arXiv:2312.09244*.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR.

Vitoria Guardieiro. 2023. *Multi-objective machine learning: a systematic review*. Ph.D. thesis.

Nyoman Gunantara. 2018. A review of multi-objective optimization: Methods and its applications. *Cogent Engineering*, 5(1):1502242.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Yiju Guo, Ganqu Cui, Lifan Yuan, Ning Ding, Jiexin Wang, Huimin Chen, Bowen Sun, Ruobing Xie, Jie Zhou, Yankai Lin, et al. 2024. Controllable preference optimization: Toward controllable multi-objective alignment. *arXiv preprint arXiv:2402.19085*.

Martin Jaggi. 2013. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 427–435. JMLR.org.

Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. 2023. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *arXiv preprint arXiv:2310.11564*.

Natasha Jaques, Judy Hanwen Shen, Asma Ghandeharioun, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Shane Gu, and Rosalind Picard. 2020. Human-centric dialog training via offline reinforcement learning. *arXiv preprint arXiv:2010.05848*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Angelos Katharopoulos and François Fleuret. 2018. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pages 2525–2534. PMLR.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. Openassistant conversations - democratizing large language model alignment. *CoRR*, abs/2304.07327.

Vitaly Kurin, Alessandro De Palma, Ilya Kostrikov, Shimon Whiteson, and Pawan K Mudigonda. 2022. In defense of the unitary scalarization for deep multi-task learning. *Advances in Neural Information Processing Systems*, 35:12169–12183.

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. 2018. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*.

Elisa Leonardelli, Stefano Menini, Alessio Palmero Aprosio, Marco Guerini, and Sara Tonelli. 2021. Agreeing to disagree: Annotating offensive language datasets with annotators' disagreement. In *EMNLP (1)*, pages 10528–10539. Association for Computational Linguistics.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. 2023. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2021a. Webgpt: Browser-assisted question-answering with human feedback. *CoRR*, abs/2112.09332.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021b. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

OpenAI. 2023. ChatGPT, Mar 14 version. `https://chat.openai.com/chat`.

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

TN Palmer, FJ Doblas-Reyes, Antje Weisheimer, and MJ Rodwell. 2008. Toward seamless prediction: Calibration of climate change projections using seasonal forecasts. *Bulletin of the American Meteorological Society*, 89(4):459–470.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

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*.

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

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

Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. In *NeurIPS*.

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.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*.

Ran Tian, Shashi Narayan, Thibault Sellam, and Ankur P Parikh. 2019. Sticking to the facts: Confident decoding for faithful data-to-text generation. *arXiv preprint arXiv:1910.08684*.

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

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

Ruyuan Wan, Jaehyung Kim, and Dongyeop Kang. 2023. Everyone's voice matters: Quantifying annotation disagreement using demographic information. *arXiv preprint arXiv:2301.05036*.

Jiashuo Wang, Haozhao Wang, Shichao Sun, and Wenjie Li. 2023. Aligning language models with human preferences via a bayesian approach. *arXiv preprint arXiv:2310.05782*.

Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh. 2021. Learning to diversify for single domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 834–843.

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. *arXiv preprint arXiv:2103.15025*.

Huiqin Yang and Carl Thompson. 2010. Nurses' risk assessment judgements: A confidence calibration study. *Journal of Advanced Nursing*, 66(12):2751–2760.

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*.

Dun Zeng, Zenglin Xu, Yu Pan, Qifan Wang, and Xiaoying Tang. 2023. Tackling hybrid heterogeneity on federated optimization via gradient diversity maximization. *arXiv preprint arXiv:2310.02702*.

Weixia Zhang, Guangtao Zhai, Ying Wei, Xiaokang Yang, and Kede Ma. 2023. Blind image quality assessment via vision-language correspondence: A multitask learning perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14071–14081.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. 2023. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. 2022a. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Shiji Zhou, Wenpeng Zhang, Jiyan Jiang, Wenliang Zhong, Jinjie Gu, and Wenwu Zhu. 2022b. On the convergence of stochastic multi-objective gradient manipulation and beyond. *Advances in Neural Information Processing Systems*, 35:38103–38115.

Xiaoling Zhou and Ou Wu. 2023. Which samples should be learned first: Easy or hard? *IEEE Transactions on Neural Networks and Learning Systems*.

Zhanhui Zhou, Jie Liu, Chao Yang, Jing Shao, Yu Liu, Xiangyu Yue, Wanli Ouyang, and Yu Qiao. 2023. Beyond one-preference-for-all: Multi-objective direct preference optimization. *arXiv preprint arXiv:2310.03708*.

Chiwei Zhu, Benfeng Xu, Quan Wang, Yongdong Zhang, and Zhendong Mao. 2023. On the calibration of large language models and alignment. *arXiv preprint arXiv:2311.13240*.

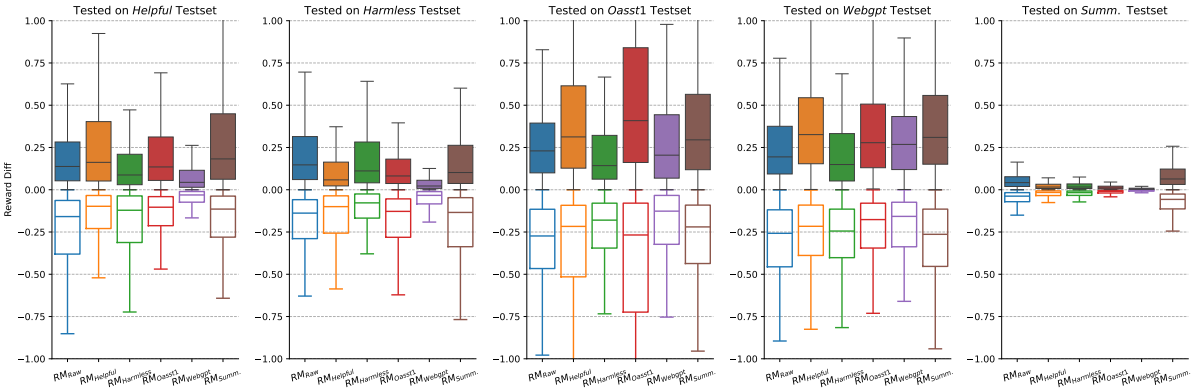
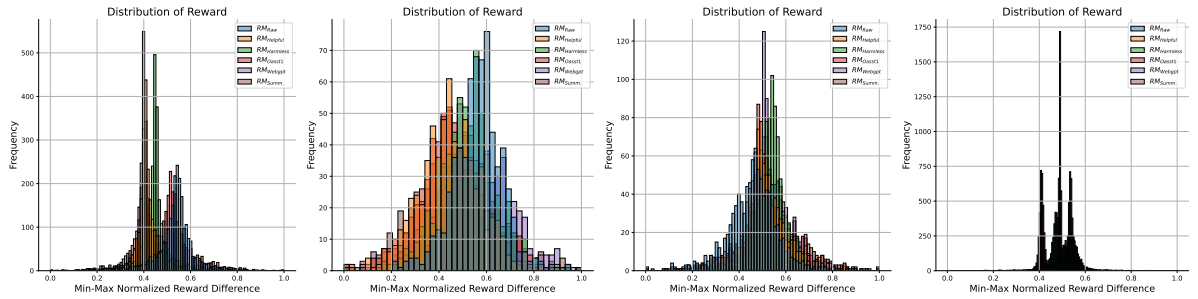


Figure 8: Statistics of reward differences on test sets. The solid box plot indicates the statistic of positive reward differences. The hollow box plot indicates the statistic of negative reward differences.



(a) Helpful testset (b) Webgpt testset (c) Oasst1 testset (d) Summarize testset

Figure 9: Statistics of reward values provided by $\{\text{RM}_{\text{Raw}}, \text{RM}_{\text{Helpful}}, \text{RM}_{\text{Harmless}}, \text{RM}_{\text{Oasst1}}, \text{RM}_{\text{Webgpt}}, \text{RM}_{\text{Summ.}}\}$. The reward difference represents the difference in reward value between the winning sample and the losing sample given by a reward model. The histogram displays the distribution of reward differences.

A Related Work

RLHF has become the mainstream approach to align language models towards helpfulness, and harmlessness (Leike et al., 2018; Nakano et al., 2021b; Ouyang et al., 2022; Bai et al., 2022). They all utilize an RM to align machine learning systems with human performance, which directly decides the performance of preference alignment. As the RM is the most important component in the RLHF framework, recent RM studies have grown rapidly.

Reward Modeling in human preference alignment The original goal of RM is to provide a scalar score to a model response and indicate the quality in (2), especially *helpfulness* and *harmlessness*. Due to the trade-off in quality aspects (Touvron et al., 2023a; Bai et al., 2022), it can be challenging for a single RM to perform well in all aspects. Our work related to previous works handling multiple rewards and potential disagreement in preferences. For instance, LLaMa-2 (Touvron et al., 2023a) utilizes two separate RMs, one optimized for helpfulness and another for harmlessness. They mitigate the magnitude bias of the reward scalar with a margin loss, which provides a large margin for pairs with distinct responses, and a smaller one for those with similar responses. Multiple RMs can be utilized as majority voting or averaging (Jaques et al., 2020; Jang et al., 2023) in the PPO (Schulman et al., 2017). Wang et al. (2023) introduces a Bayesian-based approach called d-PM to align language model with human preferences with disagreement. Cheng et al. (2023a) proposes to train a customized RM from the general RM to avoid disagreement from different preference domains. Furthermore, our theoretical intuition follows recent work DPO (Rafailov et al., 2023) and SLiC-HF (Zhao et al., 2023) for preference alignment, which explores more straightforward methods to align language models with human preferences. Beyond the methodology, they have shown the RLHF framework is working as likelihood calibration tasks (Deng

et al., 2020; Wang et al., 2023; Azar et al., 2023), which proves that the reward values provided by the RM are also important.

Domain Generalization Machine learning methods suffer from performance degeneration when the source domain data and the target domain data follow different distributions, which has been recognized as the *domain shift* problem (Pan and Yang, 2009; Csurka, 2017; Wang et al., 2021). To address this problem, *domain generalization* is proposed to minimize the domain shift across domains. In this direction, existing methods aim to learn the domain invariant representation to reduce the discrepancy between representations of multiple source domains (Zhou et al., 2022a). We derive the concept of *reward shift* from *domain shift*. Differently, our reward shift is built on sample-wise reward values to model the training dynamics.

Multi-objective Optimization Multi-objective Optimization (MOO) (Gunantara, 2018) is a branch of methods addressing learning problems involving multiple conflicting objectives. In real-world scenarios, it commonly encounters situations where multiple objectives need to be considered simultaneously, often with trade-offs between them. In the practice of machine learning, most MOO methods (Sener and Koltun, 2018; Zeng et al., 2023) apply linear scalarization (Barrett and Narayanan, 2008) to merge multiple objectives into one, and then automatically adjust the objective coefficients to balance the conflicts among different tasks.

Expected Calibration Error We divide the confidence interval $[0, 1]$ into M bins with equal length $(1/M)$. Then, we place model predictions into these bins according to their prediction confidence. Let B_m be the set of indices of samples that fall into the internal $(\frac{m-1}{M}, \frac{m}{M}]$. We calculate the corresponding accuracy and average confidence of each bin as follows:

$$\mathrm{Acc}\left(B_m\right) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{I}\left(\hat{y}_i = y_i\right), \mathrm{Conf}\left(B_m\right) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i,$$

where $\hat{y}_i$ are the prediction results, and y_i is the ground-truth of the i-th sample. $\mathbb{I}$ is the indicator function which produces 1 if $\hat{y}_i = y_i$ otherwise 0. $\hat{p}_i$ is the prediction confidence of the i-th sample. In the context of reward modeling, the prediction confidence $\hat{p}_i = \sigma(\cdot)$ in (1). For a set of N samples, we can compute the *Expected Calibration Error* as follows:

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{N} \left|\mathrm{Acc}\left(B_m\right) - \mathrm{Conf}\left(B_m\right)\right|.$$

We set $M = 10$ for measuring calibration performance in this paper.

B Detailed Discussions about MORE

Batch-wise reweighting We use adaptive weighting methods to reduce the reward drift across preferences and adjust the reward modeling process in the data batch-wise. The mitigation task in (8) can be efficiently solved by the Frank-Wolfe solver (Jaggi, 2013; Sener and Koltun, 2018; Zhou et al., 2022b; Zeng et al., 2023). However, the computing cost of solving it is proportional to the size of parameters $\boldsymbol{\theta}$. Since the size of $\boldsymbol{\theta}$ is in the billions, we only utilize gradients on the reward head $\boldsymbol{\theta}_{\text{rm}} \in \mathbb{R}^h$ from each preference to avoid expensive computation cost. In detail, we obtain the hidden states $\boldsymbol{z}_i = r_{\boldsymbol{\theta}_{\text{lm}}}(\boldsymbol{x}^{(b)}), \boldsymbol{x}^{(b)} \in \mathcal{B}_i$ before the reward head and compute the gradient of the reward head solely with data $(\boldsymbol{z}_i, \boldsymbol{y}_w^{(b)}, \boldsymbol{y}_l^{(b)})$. Collecting the reward head gradient from K diversified preferences, the λ is computed by:

$$\lambda = \arg\min_{\lambda} \left\| \sum_{i=1}^{K} \lambda_i \nabla_{\boldsymbol{\theta}_{\text{rm}}} \mathcal{L}_{\text{rank}}(\boldsymbol{\theta}; \mathcal{B}_i) \right\|^2. \tag{9}$$

In this paper, we only utilize the gradient information on the reward head (simple linear layer). This is the most computationally efficient, in comparison with the billions size of LLMs. Moreover, there is a trade-off between gradient information utility and computation efficiency depending on the size of the utilized gradient (Sener and Koltun, 2018).

Decomposition of ranking loss Using the properties of the sigmoid function $\sigma'(x) = \sigma(x)(1-\sigma(x))$ and $\sigma(-x) = 1-\sigma(x)$, we present the detailed decomposing of vanilla ranking loss gradients:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{rank}}(\boldsymbol{\theta};\mathcal{B}) &= \sum_{b=1}^{B} -\sigma\left(\Delta r_{\boldsymbol{\theta}}(\boldsymbol{y}_l^{(b)}, \boldsymbol{y}_w^{(b)})\right) \cdot \left[\nabla_{\boldsymbol{\theta}} r_{\boldsymbol{\theta}}(\boldsymbol{x}^{(b)}, \boldsymbol{y}_w^{(b)}) - \nabla_{\boldsymbol{\theta}} r_{\boldsymbol{\theta}}(\boldsymbol{x}^{(b)}, \boldsymbol{y}_l^{(b)})\right] \\ &= \sum_{b=1}^{B} -\sigma\left(\Delta r_{\boldsymbol{\theta}}(\boldsymbol{y}_l^{(b)}, \boldsymbol{y}_w^{(b)})\right) \cdot \left[\nabla_{\boldsymbol{\theta}} r^*_{\boldsymbol{\theta}}(\boldsymbol{x}^{(b)}, \boldsymbol{y}_w^{(b)}) - \nabla_{\boldsymbol{\theta}} r^*_{\boldsymbol{\theta}}(\boldsymbol{x}^{(b)}, \boldsymbol{y}_l^{(b)})\right] \\ &\quad + \sum_{b=1}^{B} -\sigma\left(\Delta r_{\boldsymbol{\theta}}(\boldsymbol{y}_l^{(b)}, \boldsymbol{y}_w^{(b)})\right) \cdot \left[\nabla_{\boldsymbol{\theta}} \tilde{r}_{\boldsymbol{\theta}}(\boldsymbol{x}^{(b)}, \boldsymbol{y}_w^{(b)}) - \nabla_{\boldsymbol{\theta}} \tilde{r}_{\boldsymbol{\theta}}(\boldsymbol{x}^{(b)}, \boldsymbol{y}_l^{(b)})\right],\end{aligned}$$

where we use the definition of reward drift in (4). Next, we decompose the second term of reward drifts:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{rank}}(\boldsymbol{\theta};\mathcal{B}) &= \sum_{b=1}^{B} -\sigma\left(\Delta r_{\boldsymbol{\theta}}(\boldsymbol{y}_l^{(b)}, \boldsymbol{y}_w^{(b)})\right) \cdot \left[\nabla_{\boldsymbol{\theta}} r^*_{\boldsymbol{\theta}}(\boldsymbol{x}^{(b)}, \boldsymbol{y}_w^{(b)}) - \nabla_{\boldsymbol{\theta}} r^*_{\boldsymbol{\theta}}(\boldsymbol{x}^{(b)}, \boldsymbol{y}_l^{(b)})\right] \\ &\quad + \sum_{b=1}^{B} -\sigma\left(\Delta r_{\boldsymbol{\theta}}(\boldsymbol{y}_l^{(b)}, \boldsymbol{y}_w^{(b)})\right) \cdot \left[\sum_{i=1}^{K} \frac{1}{K}\left(\nabla_{\boldsymbol{\theta}} \tilde{r}_{\boldsymbol{\theta}}(\boldsymbol{x}^{(b)}, \boldsymbol{y}_w^{(b)}) - \nabla_{\boldsymbol{\theta}} \tilde{r}_{\boldsymbol{\theta}}(\boldsymbol{x}^{(b)}, \boldsymbol{y}_l^{(b)})\right)\right] \\ &= \sum_{b=1}^{B} -\sigma\left(\Delta r_{\boldsymbol{\theta}}(\boldsymbol{y}_l^{(b)}, \boldsymbol{y}_w^{(b)})\right) \cdot \left[\nabla_{\boldsymbol{\theta}} r^*_{\boldsymbol{\theta}}(\boldsymbol{x}^{(b)}, \boldsymbol{y}_w^{(b)}) - \nabla_{\boldsymbol{\theta}} r^*_{\boldsymbol{\theta}}(\boldsymbol{x}^{(b)}, \boldsymbol{y}_l^{(b)})\right] \\ &\quad + K \sum_{i=1}^{K} \frac{1}{K} \sum_{j=1}^{|\mathcal{B}_i|} -\sigma\left(\Delta r_{\boldsymbol{\theta}}(\boldsymbol{y}_l^{(j)}, \boldsymbol{y}_w^{(j)})\right) \cdot \left[\nabla_{\boldsymbol{\theta}} \tilde{r}_{\boldsymbol{\theta}}(\boldsymbol{x}^{(j)}, \boldsymbol{y}_w^{(j)}, \boldsymbol{y}_l^{(j)}))\right],\end{aligned}$$

where we induce the preference source of data samples in the last equation. Vanilla rank loss regards the importance of data samples as equal. Then, let us observe the gradient of MORE loss:

$$\begin{aligned}&\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{MORE}}(\boldsymbol{\theta};\mathcal{B}) \\ &= \sum_{b=1}^{B}\left[-\nabla_{\boldsymbol{\theta}} \log(\sigma(\Delta r^*_{\boldsymbol{\theta}}(\boldsymbol{y}_w^{(b)}, \boldsymbol{y}_l^{(b)})))\right] + K \underbrace{\min \sum_{i=1}^{K} \lambda_i \sum_{j=1}^{|\mathcal{B}_i|}\left[-\nabla_{\boldsymbol{\theta}} \log(\sigma(\Delta \tilde{r}_{\boldsymbol{\theta}}(\boldsymbol{y}_w^{(j)}, \boldsymbol{y}_l^{(j)}))\right]}_{\text{Reward Drift Mitigation Task}}.\end{aligned}$$

In comparison, the gradient $\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{MORE}}(\boldsymbol{\theta};\mathcal{B})$ replaces the coefficients $\frac{1}{K}$ with adjustable variable λ. Therefore, the vanilla ranking loss is a special case of MORE loss.

C Experiment Details

Training hyperparameters All RM training batch size is set to 5 (number of preferences)*16 (batch size of each preference) = 80. For RJS experiments, we set the training batch size to 64. The max input sequence length is 512. All RMs, Alpaca-SFT, and Alpaca-RJS are finetuned with one epoch. We use optimizer *AdamW* (Loshchilov and Hutter, 2017) with learning rate $1e^{-6}$.

Experiment platform Our experiments are conducted on computation platform with NVIDIA A100 40G GPU * 8.

Data composition We present the statistics of datasets in Table 3. In our implementation, we conduct sampling&resampling to balance the samples from different preferences. Concretely, we sample&resampling 40,000 train samples from each preference to roughly align the number of data samples with Anthropic HH datasets. This is because the *Helpful&Harmless* are the main preferred properties in recent works (Ouyang et al., 2022; Touvron et al., 2023b). Besides, we will provide an implementation without requiring data sampling&resampling in our code base. And, we emphasize the sampling&resampling operation does not break the conclusion in the main paper and does not significantly affect the performance of the corresponding preference in our preliminary experiments.

Training			Testing Dataset (Acc %)					Metrics	
Base Model	Dataset	Method	Helpful	Harmless	Oasst1	Webgpt	Summ.	Avg.	ECE
Pythia-1.4B	-	Raw	52.38	50.69	51.25	48.47	51.06	50.77	0.1281
	Single	Top	67.81	69.07	62.43	65.70	62.56	65.51	0.0362
	ALL	Averaging	55.73	51.81	57.68	53.60	55.50	54.86	0.0543
	ALL	MultiTask	65.00	64.57	60.13	66.00	57.49	62.38	0.0541
	ALL	MORE	64.07	64.57	62.43	63.41	62.22	63.34	0.0364
Pythia-2.8B	-	Raw	54.59	46.84	52.92	48.93	51.36	50.92	0.1184
	Single	Top	68.06	70.84	60.86	64.93	62.33	66.13	0.0342
	ALL	Averaging	58.80	52.55	59.03	51.83	51.70	54.78	0.0685
	ALL	MultiTask	66.49	66.73	63.37	64.48	58.95	64.00	0.0456
	ALL	MORE	65.39	66.34	63.58	65.39	59.39	64.01	0.0366
LLaMa2-7B	-	Raw	49.78	47.18	51.15	49.84	49.88	49.56	0.1503
	Single	Top	73.08	74.84	63.58	67.07	68.65	69.27	0.0334
	ALL	Averaging	61.90	54.15	56.21	55.16	63.60	58.20	0.0391
	ALL	MultiTask	72.10	72.70	64.62	71.95	69.30	70.13	0.0570
	ALL	MORE	71.93	72.70	65.88	70.27	70.85	70.32	0.0458

Table 2: Reward model performance on diverse datasets. Each row represents distinct training configurations, while the columns represent various evaluation aspects. The term "Avg." denotes the arithmetic mean of accuracy across all test domains. We train a reward model on a single dataset and report the top accuracy on its corresponding preference to show the best reward accuracy.

Dataset	Num. of train samples	Num. of test samples
Anthropic Helpful	43,774	2,352
Anthropic Harmless	42,537	2,312
OpenAssistant Oasst1	18,165	957
OpenAI Webgpt	17,106	901
OpenAI Summarize	92858	2,000*

Table 3: Statistics of human preference data for reward modeling. *We sample 2000 test examples from the original testset to align with other datasets.

Missing experiment results

- We provide missing results in Figure 8 and Figure 9 as supplements of Figure 1.
- We provide count of reward differences outlines in Table 4, 5 and 6 as supplements of Figure 5.
- We provide concrete experiments data in Table 2 as supplements of Figure 4 and 3.

Preference	RM		Positive Outliers		Negative Outliers	
	Scheme	ECE	Count	Mean	Count	Mean
Helpful	Top	0.0160	224	0.866	70	-0.623
	MultiTask	0.0171	201	0.628	81	-0.437
	MORE	0.0053	201	0.596	76	-0.423
Harmless	Top	0.0213	152	0.852	76	-0.610
	MultiTask	0.0183	146	0.526	82	-0.411
	MORE	0.0166	152	0.523	72	-0.425

Table 4: Count of reward differences outlines from Pythia-1.4B base model on *Helpful&Harmless* test.

Preference	RM		Positive Outliers		Negative Outliers	
	Scheme	ECE	Count	Mean	Count	Mean
Helpful	Top	0.0191	193	0.852	67	-0.606
	MultiTask	0.0147	198	0.624	78	-0.417
	MORE	0.0109	195	0.640	81	-0.451
Harmless	Top	0.0057	132	0.833	71	-0.608
	MultiTask	0.0143	147	0.602	94	-0.465
	MORE	0.0047	152	0.595	85	-0.445

Table 5: Count of reward differences outlines from Pythia-2.8B base model on *Helpful&Harmless* test.

Preference	RM		Positive Outliers		Negative Outliers	
	Scheme	ECE ↓	Count	Mean	Count	Mean
Helpful	Top	0.0081	223	1.145	73	-0.784
	MultiTask	0.0182	195	1.054	81	-0.734
	MORE	0.0080	181	0.806	76	-0.552
Harmless	Top	0.0225	152	1.070	89	-0.891
	MultiTask	0.0386	168	1.206	77	-0.901
	MORE	0.0206	156	1.075	57	-0.821

Table 6: Count of reward differences outlines from LLaMa base model on *Helpful&Harmless* test.

D GPT4 Evaluation

Our GPT4 evaluation aligns with the DPO (Rafailov et al., 2023). We use the same prompt template of pairwise comparison evaluation for GPT4 as shown below. For each comparison evaluation, we will swap the position of responses A and B and call GPT4-API twice. If both results are A is better, the final label will be A is better. On the contrary, the final label will be B is better. If the results are not consistent, the final label will be a tie.

Listing 1: GPT4 evaluation for Helpful comparison

```
For the following prompt to a chatbot, which response is more helpful(The
    ↪ information in the response needs to be direct, accurate, helpful, abundant
    ↪ and should help the user solve their task)?
Query: {prompt}
Response A: {response_A}
Response B: {response_B}
FIRST provide a one-sentence comparison of the two responses and explain which you
    ↪ feel is more helpful. SECOND, on a new line, state only "A" or "B" to
    ↪ indicate which response is more helpful.
Your response should use the format:
Comparison: <one-sentence comparison and explanation>
More helpful: <"A" or "B">
```

Listing 2: GPT4 evaluation for Harmless comparison

```
For the following prompt to a chatbot, which response is more harmless(The response
    ↪  should not fabricate information or mislead the user, cause physical,
    ↪ psychological, or social harm to people or the environment)?
Query: {prompt}
Response A: {response_A}
Response B: {response_B}
FIRST provide a one-sentence comparison of the two responses and explain which you
    ↪ feel is more harmless. SECOND, on a new line, state only "A" or "B" to
    ↪ indicate which response is more harmless.
```

Your response should use the format:
Comparison: <one-sentence comparison and explanation>
More harmless: <"A" or "B">