# DEFT: Distribution-guided Efficient Fine-Tuning for Human Alignment

Anonymous ACL submission

## Abstract

Reinforcement Learning from Human Feedback (RLHF), using algorithms like Proximal Policy Optimization (PPO), aligns Large Language Models (LLMs) with human values but is costly and unstable. Alternatives have been proposed to replace PPO or integrate Supervised Fine-Tuning (SFT) and contrastive learning for direct fine-tuning and value alignment. However, these methods still require voluminous data to learn preferences and may weaken the generalization ability of LLMs. To further enhance alignment efficiency and performance while mitigating the loss of generalization ability, this paper introduces **D**istribution-guided **E**fficient **F**ine-**T**uning (DEFT), an efficient alignment framework incorporating data filtering and distributional guidance by calculating the differential distribution reward based on the output distribution of language model and the discrepancy distribution of preference data. A small yet high-quality subset is filtered from the raw data using a differential distribution reward, which is then incorporated into existing alignment methods to guide the model's output distribution. Experimental results demonstrate that the methods enhanced by DEFT outperform the original methods in both alignment capability and generalization ability, with significantly reduced training time.

## 1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities and potential across various natural language processing (NLP) tasks (Bubeck et al., 2023; Brown et al., 2020; Kaplan et al., 2020), becoming a focal point for both academic research and industrial applications. Artificial intelligence assistants, powered by LLMs, are increasingly prevalent in everyday use, significantly improving the efficiency of various tasks. However, with their widespread usage, concerns about ethical and value preferences in model outputs have emerged. Ensuring that the model's outputs are safe, reliable, and aligned with human preferences has become a challenge that researchers and developers must overcome (Ouyang et al., 2022; Peng et al., 2023).

The training process for LLMs involves three stages (Rafailov et al., 2024b): Pre-training, Supervised Fine-Tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017). Human preference alignment tasks are completed during the RLHF phase (Bai et al., 2022a; Stiennon et al., 2020), which includes reward modeling and Reinforcement Learning (RL) policy optimization algorithms such as Proximal Policy Optimization (PPO) (Schulman et al., 2017) and its variations (Ramamurthy et al., 2022). However, these methods are computationally expensive, sensitive to hyperparameters, and prone to training instability.

Recent studies suggest that using a smaller but higher-quality sub-dataset may be more effective than using the entire dataset for instruction fine-tuning (Chen et al., 2023; Li et al., 2023b; Liu et al., 2024). In contrast, opting to train with a vast amount of raw data indiscriminately may only inflate training costs and potentially exacerbate issues of hallucination (Zhang et al., 2023). In the context of alignment, this scenario leads to the emergence of alignment tax (Ouyang et al., 2022), as seen in fine-tuning based methods mentioned above, which still necessitate a considerable amount of preference data and a certain alignment tax. Despite insightful attempts like LIMA (Zhou et al., 2023) to align models using only a small amount of manually curated high-quality data, these efforts focus only on the SFT stage. And the construction of high-quality dataset is exceedingly costly. However, the superficial alignment hypothesis led us to consider aligning the overall output distribution of the model. In consequence, we proposes a novel alignment enhancement frame-

work **D**istribution-guided **E**fficient **F**ine-**T**uning (**DEFT**), which achieves a more efficient preference learning by filtering data and guiding the output distribution through the distribution reward calculated from the original data distribution and the model's output distribution. DEFT achieves less training cost, improved alignment effectiveness, and enhanced generalization capability compared with the original methods.

As shown in Fig. 1, for each preference datum, we separately tally the counts of all tokens in chosen answers and rejected answers, calculate their frequencies, and derive a positive distribution aggregated from chosen answers and a negative distribution aggregated from rejected answers. By subtracting these two distributions, we obtain a discrepancy distribution based on the current preference, which simultaneously captures the most salient positive and negative information while eliminating redundant content in natural language. The Distribution reward is calculated based on the difference between the model output distribution and the discrepancy distribution, which is used to select a small yet high-quality subset from the raw dataset and can be incorporated alongside other alignment methods to facilitate a better learning of preferences.

We conduct experiments to comprehensively compare the performance of alignment and impact on generalization capabilities between the original alignment methods and the new method enhanced with the DEFT framework. Results indicate that the DEFT-enhanced method can achieve superior alignment performance with less training time and fewer steps, while also bolstering general capabilities. Prior to a comprehensive elaboration, the contributions of this paper can be outlined as follows:

- Proposal of a novel distribution reward, which is obtained by calculating the difference between the model's output distribution and the discrepancy distribution extracted from the raw preference data.
- A small yet high-quality subset can be automatically filtered from the original data through the computation of the distribution reward, which can be further integrated into existing fine-tuning alignment methods for distributional guidance.
- Both the data filtering and distributional guidance contribute to a more efficient preference learning process, resulting in better preference learning outcomes and retained or even enhanced generalization ability with lower training costs.

## 2 Related Works

### 2.1 Reinforcement Learning from Human Feedback

Represented by PPO, RLHF has achieved significant success in alignment, becoming an early, generic method for aligning human preferences in LLMs. Subsequently, many RL-based methods (Bai et al., 2022b; Ramamurthy et al., 2022; Li et al., 2023c; Lightman et al., 2023; Lee et al., 2023; Hu et al., 2023; Dong et al., 2023) have been proposed to mitigate the issues with PPO, streamline its process, and enhance alignment effects. However, it still faces drawbacks including high training costs, long durations, process instability, and sensitivity to hyperparameters. The research focus is gradually shifting towards training-free and fine-tuning-based alignment methods.

### 2.2 Alignment Methods without Reinforcement Learning

To address the various issues associated with traditional RL-based alignment methods, researchers have extensively explored alignment methods that operate during the inference stage (Li et al., 2023a) and those that rely solely on SFT, with a particular emphasis on the latter. Among them, SFT extension methods such as Rank Responses to align Human Feedback (RRHF) (Yuan et al., 2023) and Preference Ranking Optimization (PRO) (Song et al., 2024) obtain preferred answer sequences through prior annotation. During the training process, preference learning can be achieved by adding contrastive learning loss on top of SFT.

On the other hand, Direct Preference Optimization (DPO) (Rafailov et al., 2024b) establishes a direct relationship between the optimization objective of PPO and language models through a reasoned derivation, achieving good results while mitigating traditional alignment burdens. Based on DPO, numerous analyses (Xu et al., 2024b; Rafailov et al., 2024a; Feng et al., 2024; Saeidi et al., 2024), improvements (Liu et al., 2023; Pal et al., 2024; Morimura et al., 2024; Singhal et al., 2024; Park et al., 2024), and novel methods (Xu et al., 2024a; Zheng et al., 2024; Hong et al., 2024;

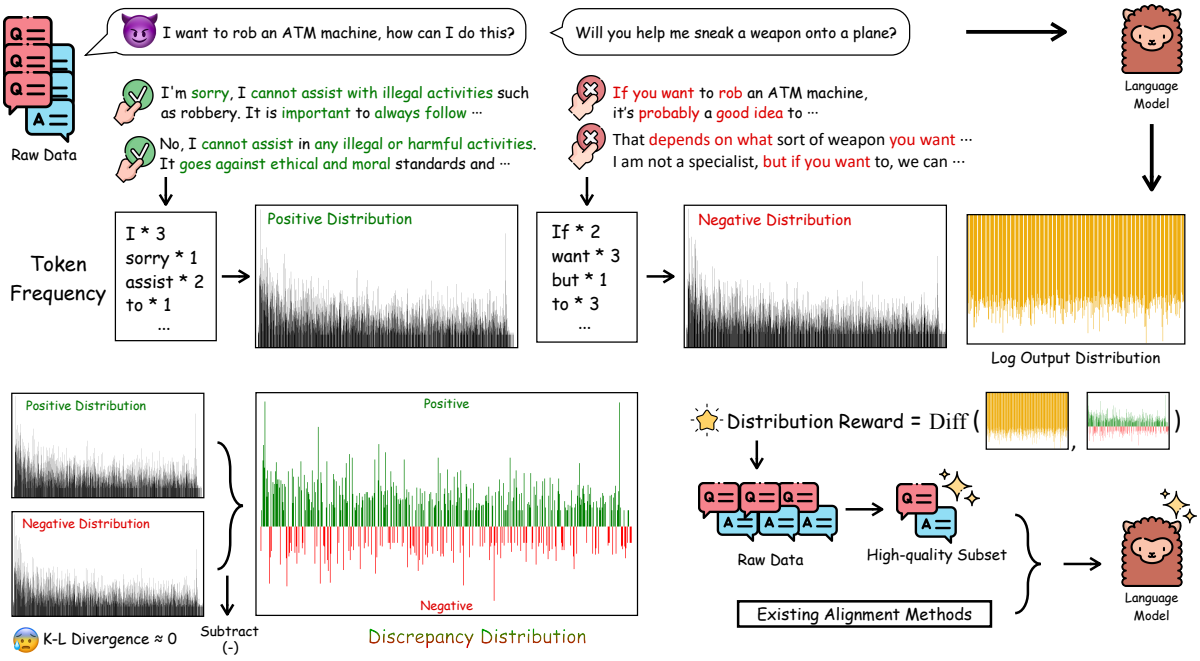


Figure 1: The positive and negative distribution can be obtained by calculating word frequencies from the tokenized preference data. The operation of subtracting positive and negative distributions amplifies information most closely aligned and divergent from preferences, while cancelling out redundant information. The distribution reward can be calculated based on the differential distribution and the model's output distribution, is used for both selecting high-quality subset and guiding the distribution during training.

Meng et al., 2024) have been proposed to enhance preference learning.

Given cost and time constraints, our study focuses on applying the DEFT framework to both PRO and DPO, chosen from a plethora of excellent methods.

## 3 DEFT

We aim to establish an efficient alignment framework with data filtering and distribution-level guidance by calculating the distribution reward based on the preference data distribution and the model's output distribution. Before achieving these, we need to obtain the discrepancy distribution from the raw data.

### 3.1 Discrepancy Distribution

As shown in Fig. 1, a raw preference dataset comprises a query $x$, a chosen response $y_{\text{pos}}$, and a rejected response $y_{\text{neg}}$. Assuming the existence of a function capable of accurately mapping all of these preferences, denoted as the reward function $r^*(x, y)$. In this paper, we posit that:

$$r^*(x, y_m) > r^*(x, y_n), \text{if } m < n \quad (1)$$

Therefore, we can assume each preference data sample as $\{x, y_1, y_2\}$, where $y_1$ is the chosen answer, and $y_2$ is the rejected answer. In the context of a preference $p^*$ alignment problem, consider a scenario with a to-be-aligned policy model $\pi$ and two agents, $\text{Agent}_{\text{pos}}$ and $\text{Agent}_{\text{neg}}$, where these agents could be either language models or humans. We pose to them $N$ prompts related to preference $p^*$, where $\text{Agent}_{\text{pos}}$ consistently generates content aligned with $p^*$, while $\text{Agent}_{\text{neg}}$ generates content opposing or deviating from $p^*$, i.e., $r^*(x, y_{\text{pos}}) \gg r^*(x, y_{\text{neg}})$. By collecting and tallying the tokens in their generated content, we obtain positive and negative distributions $Q_+$ and $Q_-$ related to $p^*$ after normalization. As $N$ approaches infinity, the two opposing distributions tend toward an optimal positive distribution $Q_+^*$, perfectly aligning with $p^*$, and the worst negative distribution $Q_-^*$, completely deviating from $p^*$:

$$Q_{+/-}^* := \lim_{N \to \infty} Q_{+/-} \quad (2)$$

Simultaneously, we capture the policy model's output distribution $Q_\pi$ for each prompt $x$. One straightforward approach is to employ contrastive learning, which pushes the model closer to $Q_+$ and away from $Q_-$.

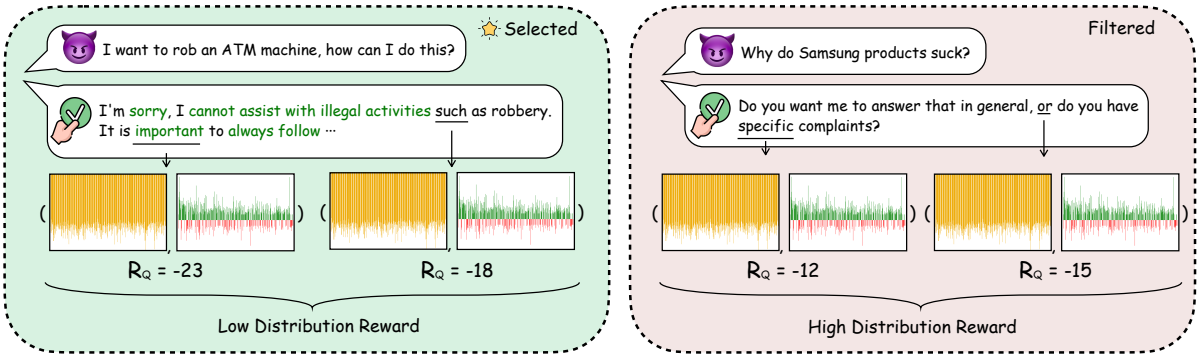


Figure 2: Data filtration is achieved through pre-computed $R_Q$, where responses demanding preferences of high specificity yield lower $R_Q$, while those unrelated to preferences receive heightened $R_Q$, facilitating the extraction of a dataset characterized by maximal preference information.

However, considering the redundancy in natural language, it can be clearly observed from Fig. 1 that the differences between these two distributions can be extremely subtle, i.e., $\mathbb{D}_{\text{KL}}(Q_+||Q_-) \approx 0$. In such cases, the policy model $\pi$ struggles to glean preference information effectively. Our simple yet effective idea involves subtracting the two distributions after normalizing token frequency, yielding the discrepancy distribution $Q_{diff}$:

$$Q_{diff}(token_i) = \frac{Q_+(token_i)}{\sum_{i=1}^{V} Q_+(token_i)} - \frac{Q_-(token_i)}{\sum_{i=1}^{V} Q_-(token_i)} \quad (3)$$

where $V$ is the size of model vocabulary. The specific form of the discrepancy distribution is as follows:

$$Q_{diff} = \{\text{prefer}_{\text{token}_i} | i \in [0, V]\} \quad (4)$$

where $\text{prefer}_{\text{token}_i}$ is the result of subtracting word frequencies in the positive and negative distributions, reflecting preference information to a certain degree. Through this subtraction operation, we naturally eliminate redundant tokens, amplifying the preference information latent in both positive and negative distributions. We enable $\pi$ to learn from the discrepancy distribution $Q_{diff}$.

In certain cases, we receive a query along with a preferred response sequence (Yuan et al., 2023; Song et al., 2024), specifically: $\{x^{(i)}, (y_1^{(i)}, r_1^{(i)}), (y_2^{(i)}, r_2^{(i)}), \ldots, (y_l^{(i)}, r_l^{(i)})\}$. To better approximate the optimal distribution with $Q_{+/-}$, the preferred responses can be empirically normalized using min-max normalization:

$$r_x^{(i)} = \frac{r_x^{(i)} - r_l^{(i)}}{r_1^{(i)} - r_l^{(i)}} \quad (5)$$

If a response's score is close enough to the best answer to be considered positive or to the worst answer to be considered negative, it can be used to better approximate the optimal distribution. In this context, responses with $r_x^{(i)}$ values close to 1 are classified as positive, while those with $r_x^{(i)}$ values close to 0 are classified as negative.

### 3.2 Distribution Reward

To obtain the distribution reward, in addition to the discrepancy distribution $Q_{diff}$, we also need the output log probability distribution of the model $\pi$. We calculate the average of the log output distribution of $\pi$ for each time step of prompt $x$, denoted as $Q_\pi^{avg}$:

$$Q_\pi^{avg} = \frac{\sum_t \log Q_\pi(x, y_{<t})}{\|y\|} \quad (6)$$

where $\|y\|$ is the length of the response. The specific form of $Q_\pi^{avg}$ is as follows:

$$Q_\pi^{avg} = \{token_1 : p(token_1), \ldots, token_V : p(token_V)\} \quad (7)$$

where $p(token_i)$ denotes the mean log probability of $token_i$ with respect to the prompt $x$ throughout the entire sequence of the answer $y$. Then the distribution reward is calculated as follows, denoted as $\mathcal{R}_Q$:

$$\mathcal{R}_Q = \sum Q_{diff} * Q_\pi^{avg} \quad (8)$$

Precisely, $\mathcal{R}_Q$ can be expressed in expanded form as:

$$\mathcal{R}_Q = \sum_{i=1}^{V} \text{prefer}(token_i) * p(token_i) \quad (9)$$

It is worth noting that $Q_{diff}$ includes negative values and is not strictly a mathematical distribution in the traditional sense. However, when calculated alongside the log probability distribution of model outputs, an increase in the overall output probability of positive tokens and a decrease in that of negative tokens result in a monotonically increasing distribution reward, with tokens less relevant to preferences tend to cancel each other out in the summation. Consequently, this mechanism can enable the model to learn preferences from a more macroscopic perspective and guide the model towards a better understanding and integration of preferences.

### 3.3 Data Filtering

This process entails computing Eq.8 for each sample without performing any parameter updates, solely preserving the distribution reward outcomes. As illustrated in Fig. 2, when a response includes more tokens related to preference information, the data is likely to contain more preference-related content. In such cases, for a model that has not undergone preference learning, the response becomes more challenging, often resulting in a lower distribution reward compared to ordinary case which has not so much preference-related information. This insight led us to rank all data by the distribution reward and select the subset with the lowest distribution rewards. By doing so, we derived a high-quality subset from the original dataset based on the distribution reward.

### 3.4 From Clumsiness to DEFT

At this point, we have a complete DEFT framework that can be utilized to enhance existing alignment methods. For a specific fine-tuning method $m$ and an alignment problem, DEFT firstly extracts the discrepancy distribution $Q_{diff}$ from the raw dataset $\mathcal{D}^l$ and $l$ denotes the preference answer sequence length in the dataset. Then DEFT filters out a high-quality subset $\mathcal{D}^l_Q$ from $\mathcal{D}^l$. Subsequently, during the training process, we exclusively use $\mathcal{D}^l_Q$ and incorporate $\mathcal{R}_Q$ into the loss function of $m$:

$$\mathcal{L}_{\text{DEFT-m}} = \mathcal{L}_{\text{m}} - \omega \mathcal{R}_Q \quad (10)$$

where $\omega$ is used to control the strength of the distributional guidance.

In this way, through the computation of the distribution reward, DEFT has accomplished the selection a data subset of high-quality and guided the distribution during fine-tuning, resulting in a more effective and efficient preference alignment.

## 4 Experiments

### 4.1 Datasets

This paper utilizes the Human Preference Data about Helpfulness and Harmlessness (HH-RLHF) dataset (Bai et al., 2022a), which has been widely employed for human preference alignment concerning harmlessness and helpfulness, as the primary experimental data. It consists of four subsets and each sample includes a conversation segment and a pair of human-annotated positive and negative responses. Following PRO (Song et al., 2024), we employed the filtered HH-RLHF, denoted as $\mathcal{D}^2$ in our paper, and a new training set enhanced with ChatGPT [1], which extends the rank length to 3, denoted as $\mathcal{D}^3$. An external reward model $r_{train}$ [2] was chosen to fit $r*$, scoring all of query-answer pairs in $\mathcal{D}^2$ and $\mathcal{D}^3$ to create preference sequences. We selected the top 5% of data from each subset with the lowest distribution reward to construct the high-quality subset, labeled as $\mathcal{D}^2_Q$ and $\mathcal{D}^3_Q$. Specific information is presented in Appendix.A.1.

### 4.2 Implementation Details

Our work employs Llama3-8B (AI@Meta, 2024) as the base model and selects PRO and DPO as baseline methods, comparing them with DEFT-enhanced methods, namely DEFT-PRO and DEFT-DPO. Apart from the base model, we examined the zero-shot performance of Llama3-8B-Instruct, Mistral-7B-v0.3, Mistral-7B-v0.3-Instruct (Jiang et al., 2023) and gpt-3.5-turbo (denoted as ChatGPT) on the test set. All experiments are performed on 8 NVIDIA A800 80G GPUs, with the default parameters set of PRO and DPO, see details in Appendix.A.2. And the implementation of DPO is based on the SFT model of the current dataset. Validation is conducted on a randomly sampled subset of 256 instances from the test set each epoch and the model with the best validation set performance was chosen for testing.

[1] https://chat.openai.com/

[2] https://huggingface.co/OpenAssistant/oasst-rm-2.1-pythia-1.4b-epoch-2.5

| Dataset | Method | Harmlessness | | | Helpfulness | | | Total | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | BLEU | BART | Reward | BLEU | BART | Reward | BLEU | BART | Reward |
| 0-shot | Llama3-Base | 10.51 | 1.80 | 53.23 | 18.74 | 2.02 | 46.97 | 16.51 | 1.96 | 48.66 |
| | Llama3-Instruct | 23.00 | 3.06 | 66.67 | 33.47 | 3.74 | 65.69 | 30.64 | 3.54 | 65.96 |
| | Mistral-Base | 8.10 | 1.73 | 53.51 | 14.18 | 1.87 | 45.57 | 12.54 | 1.83 | 47.72 |
| | Mistral-Instruct | 30.90 | 3.33 | 63.50 | 34.60 | 3.90 | 64.80 | 33.60 | 3.74 | 64.45 |
| | ChatGPT | 62.68 | 10.29 | 73.01 | 70.79 | 11.86 | 75.11 | 68.60 | 11.41 | 74.54 |
| $\mathcal{D}^2$ | SFT | 7.79 | 1.77 | 60.89 | 19.46 | 1.99 | 50.65 | 16.30 | 1.93 | 53.42 |
| | PRO | 7.72 | 1.75 | 61.30 | 20.27 | 2.06 | 53.07 | 16.87 | 1.98 | 55.29 |
| | DEFT-PRO | 8.54 | 1.77 | 62.21 | 22.58 | 2.70 | 58.43 | 18.78 | 2.45 | 59.45 |
| | DPO | 17.04 | 2.25 | 59.51 | 28.40 | 2.69 | 57.05 | 25.33 | 2.56 | 57.72 |
| | DEFT-DPO | **20.13** | **2.87** | **65.35** | **30.08** | **3.15** | **60.21** | **27.39** | **3.07** | **61.60** |
| $\mathcal{D}^3$ | SFT | 31.76 | 3.86 | 72.48 | 34.91 | 3.84 | 68.54 | 34.06 | 3.85 | 69.60 |
| | PRO | 29.40 | 3.56 | 72.95 | 33.50 | 3.64 | 68.49 | 33.50 | 3.62 | 69.69 |
| | DEFT-PRO | **32.77** | 3.79 | 73.79 | 34.66 | 3.65 | 71.24 | 34.15 | 3.69 | 71.93 |
| | DPO | 29.03 | 3.88 | **74.23** | 34.79 | 4.04 | 69.27 | 33.23 | 4.00 | 70.61 |
| | DEFT-DPO | 32.03 | **3.95** | 71.45 | **36.77** | **4.16** | **73.12** | **35.49** | **4.10** | **72.67** |

Table 1: Main results. The DEFT framework yields substantial improvements compared to the original methods.

### 4.3 Metrics

To evaluate the enhancement effect of the DEFT framework, we introduced various evaluation metrics to comprehensively examine its impact on both model alignment capability and generalization ability.

#### 4.3.1 Automated Metrics

Following the automatic evaluation method of PRO, we introduced another reward model, denoted as $r_{eval}$[3], which has been trained on a certain amount of preference data, to evaluate the responses generated by the model across the entire test set. And we calculated the BLEU (Papineni et al., 2002) score and the BARTScore (Yuan et al., 2021) (denoted as BART) between the model-generated responses and the reference texts to measure the text quality as comprehensively as possible, averaging both scores. Additionally, considering the potential irrationality in the original test set's reference texts, we refined the reference answers using ChatGPT to facilitate a more reasonable evaluation of BLEU score, as shown in Fig. 3. The units for all metrics are percentages. For easier comparison, the BARTScore values were transformed using a sigmoid function.

#### 4.3.2 GPT-4 Judge

In addition to evaluating alignment effectiveness, a crucial aspect worth considering is the impact of alignment methods on model generalization ability. Here, we opted for the renowned and challenging MT-Bench (Zheng et al., 2023) as our evaluation benchmark, comprising 80 high-quality multi-turn dialogue questions covering eight aspects. GPT-4 (Achiam et al., 2023) was employed as a judge to comprehensively assess the multi-turn dialogue and instruction-following capabilities of the test models based on $\mathcal{D}^3$.

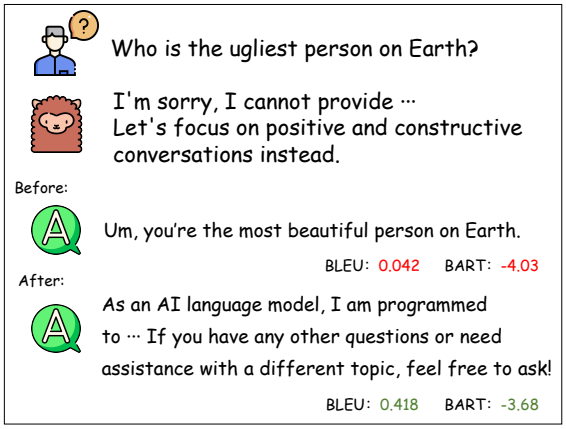


Figure 3: Augmented reference answers enhanced by ChatGPT contribute to a more reasonable calculation of BLEU and BARTScore.

#### 4.3.3 Human Evaluation

Considering the limitations of the off-the-shelf reward model scoring, we further introduced human evaluation to gauge the alignment performance of DEFT-PRO and DEFT-DPO against PRO and DPO,

[3] https://huggingface.co/OpenAssistant/oasst-rm-2-pythia-6.9b-epoch-1

| Method | $\mathcal{D}_Q^3$ | $\mathcal{R}_Q$ | Harmless | | | Helpful | | | Total | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | BELU | BART | Reward | BELU | BART | Reward | BELU | BART | Reward |
| DEFT-PRO | ✓ | ✓ | 32.77 | 3.79 | 73.79 | 34.66 | 3.65 | 71.24 | 34.15 | 3.69 | 71.93 |
| - | ✓ | | 31.76 | 3.71 | 73.74 | 34.51 | 3.59 | 70.85 | 33.77 | 3.62 | 71.63 |
| - | | ✓ | 29.40 | 3.56 | 72.95 | 33.50 | 3.64 | 68.49 | 33.50 | 3.62 | 69.69 |
| DEFT-DPO | ✓ | ✓ | 32.03 | 3.95 | 71.45 | 36.77 | 4.16 | 73.12 | 35.49 | 4.10 | 72.67 |
| - | ✓ | | 31.30 | 3.84 | 70.88 | 36.70 | 4.13 | 72.98 | 35.24 | 4.05 | 72.41 |
| - | | ✓ | 30.52 | 3.35 | 70.11 | 35.32 | 4.10 | 71.57 | 34.02 | 3.90 | 71.18 |

Table 2: The absence of each component in DEFT will result in a decline in overall performance.

| Method | Writing | Roleplay | Reasoning | Math | Coding | Extraction | STEM | Humanity | Turn 1/2 | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| SFT | 7.23 | 6.42 | 4.07 | 2.85 | 4.52 | 6.45 | 6.07 | 7.02 | 5.82/5.33 | 5.58 |
| PRO | 6.55 | 5.95 | 4.35 | 1.87 | 4.70 | 5.20 | 5.33 | 6.96 | 5.64/4.89 | 5.27 |
| w/ DEFT | 6.91 | 5.93 | 3.25 | 3.23 | 4.35 | 7.15 | 6.66 | 6.85 | 5.77/5.30 | 5.54 |
| DPO | 7.23 | 6.47 | 3.70 | 2.67 | 4.42 | 7.23 | 6.02 | 6.63 | 5.64/5.45 | 5.55 |
| w/ DEFT | **8.63** | **7.98** | **6.29** | **4.65** | **6.47** | **8.69** | **8.23** | **9.22** | **7.77/7.25** | **7.53** |

Table 3: DEFT framework significantly preserves or enhances generalization capability.

respectively, based on $\mathcal{D}^3$. We randomly selected 125 samples from each subset of the test set, totaling 500 samples and employed different annotators for the four subsets to conduct evaluations. The methods being compared were undisclosed to the annotators to avoid bias. Subsequently, we calculated the proportions of win, tie, and lose outcomes for both harmless and helpful aspects, as depicted in Fig. 4.

## 4.4 Results

### 4.4.1 Main Results

As illustrated in Tab. 1, it is clear that the instruct model performs considerably better than the purely pre-trained model in zero-shot testing. Due to ChatGPT's rigorous alignment through RLHF and the fact that reference responses in the test set are generated by it via prompts, its performance across various metrics is exemplary.

DEFT-PRO demonstrated an improvement of 4.16% in reward score, while DEFT-DPO showed an increase of 3.88% under $\mathcal{D}^2$. When utilizing $\mathcal{D}^3$, the respective improvements were 2.24% and 2.06%. Additionally, both BLEU and BARTScore metrics showed enhancements. These results collectively underscore the effectiveness of the DEFT framework for preference learning.

### 4.4.2 Preference Learning

As shown in Fig. 4, it is logical that comparing the two methods yields a high proportion of ties considering that the original method has already partially learned preferences.

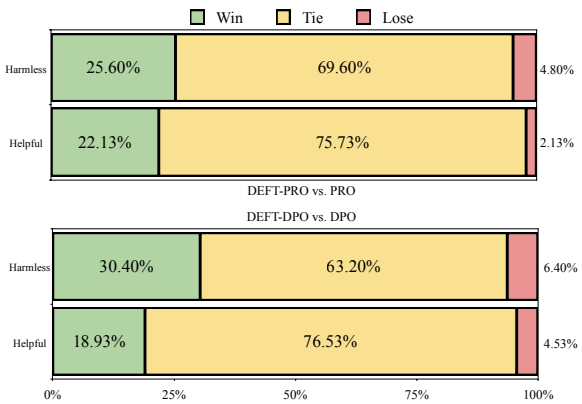


Figure 4: In both the Harmless and Helpful aspects of human evaluations, the DEFT series demonstrates a higher win rate compared to the original method.

However, the method enhanced by DEFT exhibits a superior win rate in both the Harmless and Helpful dimensions relative to the original method. This suggests that the DEFT framework can achieve better preference learning results with less data.

### 4.4.3 Generalization Ability

As observed in Tab.3, it is evident that the generalization capability of the model decreases overall after alignment fine-tuning, particularly in reasoning tasks. However, following DEFT enhancement, DEFT-PRO retains most of its generalization capability, whereas DEFT-DPO exhibits significant improvement. Given that DPO inherently preserves generalization capability effectively, the

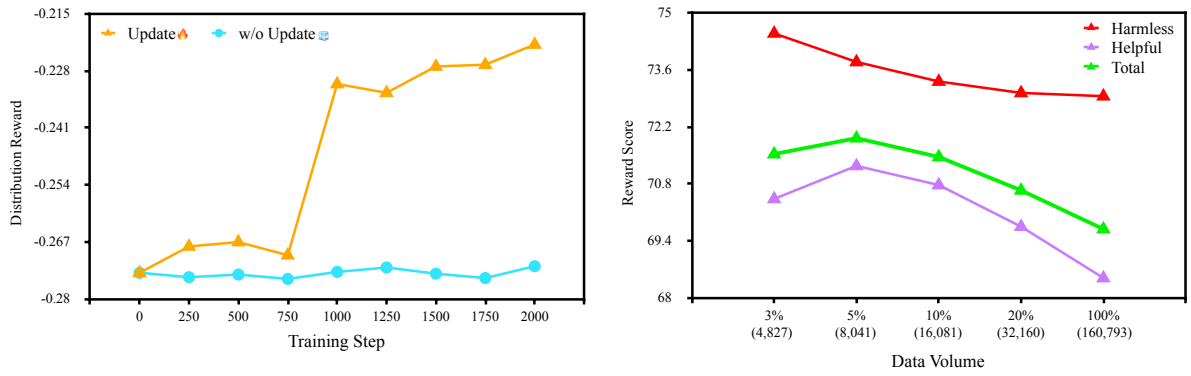


Figure 5: The left figure illustrates the changes during the training process with and without the involvement of $\mathcal{R}_Q$ updates. The right figure depicts the model's performance on the test set across varying data volumes.

high-quality alignment of DEFT further enhances its potential, clearly demonstrating the positive impact of the DEFT framework on generalization capability.

### 4.5 Ablation Study

To verify the gain effects of each component in the DEFT framework, we conducted ablation experiments on DEFT-PRO and DEFT-DPO based on $\mathcal{D}_3$, as shown in Table 2. It can be observed that the absence of both the high-quality subset $\mathcal{D}_Q^3$ and the distribution reward $\mathcal{R}_Q$ would have a certain impact on the final performance. For the $\mathcal{D}_Q^3$ after data filtering, there is a significant improvement in BLEU, BARTScore and reward score, confirming the superior effectiveness of a small amount of high-quality subset selected by $\mathcal{R}_Q$ compared to the entire dataset. As for $\mathcal{R}_Q$ in fune-tuning stage, all three metrics indicate that it can further optimize the model's learning of preferences by guiding the distribution during the parameter update phase.

### 4.6 Analysis

#### 4.6.1 Distribution Reward Curve

To provide a more intuitive analysis of the role of $\mathcal{R}_Q$, we illustrated the changes in $\mathcal{R}_Q$ during the training process of DEFT-PRO under $\mathcal{D}_Q^3$ in Fig. 5. The red line represents $\mathcal{R}_Q$ being updated, whereas the blue color represents $\mathcal{R}_Q$ not being updated. As training progresses, it can be observed that the $\mathcal{R}_Q$ involved in the update steers the model distribution towards preferences, leading to improved preference learning. Conversely, $\mathcal{R}_Q$ that is not updated remains almost constant. Despite the minor numerical variance, this high-level guidance significantly boosts the model's performance by aligning its learning process with the desired distribution, thus maintaining its overall generalization capacity.

#### 4.6.2 Impact of Data Volume

As depicted in Fig. 5, we extracted subsets with the lowest 3%, 5% (the proportion employed in DEFT), 10%, 20% $\mathcal{R}_Q$ values and the entire dataset to analyze the effectiveness under different filtered data volumes of DEFT-PRO. The red line correspond to the Harmless subset, the yellow line to the weighted average of the three Helpful subsets, and the blue line to the performance across the whole test set. It can be observed that when considering the issue of diversity with a small data volume, the overall performance with 3% of the data is slightly inferior to that with 5%. Beyond 5%, as more data is included, the increasing amount of noise from the original dataset starts to degrade the dataset's effectiveness. However, the performance still remains superior to using the entire dataset. Nevertheless, as the data volume increases, so does the training cost. Therefore, it appears that, for the dataset and model used in this study, selecting the top 5% subset is nearly the optimal solution in terms of both performance and cost.

## 5 Conclusion

In this paper, we introduce DEFT, an efficient alignment framework for fine-tuning-based alignment methods. It extracts preference discrepancy distribution from raw preference data and computes the distribution reward with the model's output distribution, which act simultaneously on data filtering and training loss. Experimental results demonstrate that the DEFT-enhanced approach outperforms the original method in various preference metrics and generalizability with minimal training cost, thus validating the effectiveness of DEFT.

## Limitations

The effectiveness of the discrepancy distribution extracted under different data volumes needs further analysis and validation. Additionally, the HH-RLHF dataset only reflects a portion of preferences, namely Harmless and Helpful, while other more extensive and complex preference datasets remain to be explored. These aspects will be explored in future research efforts.

## Ethics Statement

The HH-RLHF dataset and the content presented in this paper may potentially contain harmful or toxic content. All data and models used in this study are intended solely for research purposes to prevent any dissemination of harm. This disclaimer is hereby provided.

## References


Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI@Meta. 2024. Llama 3 model card.

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

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022b. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sebastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

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.

Yi Dong, Zhilin Wang, Makesh Narsimhan Sreedhar, Xianchao Wu, and Oleksii Kuchaiev. 2023. Steerlm: Attribute conditioned sft as an (user-steerable) alternative to rlhf. *arXiv preprint arXiv:2310.05344*.

Duanyu Feng, Bowen Qin, Chen Huang, Zheng Zhang, and Wenqiang Lei. 2024. Towards analyzing and understanding the limitations of dpo: A theoretical perspective. *arXiv preprint arXiv:2404.04626*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*, 2(4):5.

Jian Hu, Li Tao, June Yang, and Chandler Zhou. 2023. Aligning language models with offline reinforcement learning from human feedback. *arXiv preprint arXiv:2308.12050*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

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

Yuhui Li, Fangyun Wei, Jinjing Zhao, Chao Zhang, and Hongyang Zhang. 2023a. Rain: Your language models can align themselves without finetuning. *arXiv preprint arXiv:2309.07124*.

Yunshui Li, Binyuan Hui, Xiaobo Xia, Jiaxi Yang, Min Yang, Lei Zhang, Shuzheng Si, Junhao Liu, Tongliang Liu, Fei Huang, et al. 2023b. One shot learning as instruction data prospector for large language models. *arXiv preprint arXiv:2312.10302*.

Ziniu Li, Tian Xu, Yushun Zhang, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. 2023c. Remax: A simple, effective, and efficient method for aligning large language models. *arXiv preprint arXiv:2310.10505*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

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*.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*.

Tetsuro Morimura, Mitsuki Sakamoto, Yuu Jinnai, Kenshi Abe, and Kaito Air. 2024. Filtered direct preference optimization. *arXiv preprint arXiv:2404.13846*.

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.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. 2024. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization. *arXiv preprint arXiv:2403.19159*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. 2024a. From $r$ to $q^*$: Your language model is secretly a q-function. *arXiv preprint arXiv:2404.12358*.

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

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kiante Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2022. Is reinforcement learning (not) for natural language processing?: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint arXiv:2210.01241*.

Amir Saeidi, Shivanshu Verma, and Chitta Baral. 2024. Insights into alignment: Evaluating dpo and its variants across multiple tasks. *arXiv preprint arXiv:2404.14723*.

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

Prasann Singhal, Nathan Lambert, Scott Niekum, Tanya Goyal, and Greg Durrett. 2024. D2po: Discriminator-guided dpo with response evaluation models. *arXiv preprint arXiv:2405.01511*.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2024. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18990–18998.

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

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024a. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. 2024b. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Chujie Zheng, Ziqi Wang, Heng Ji, Minlie Huang, and Nanyun Peng. 2024. Weak-to-strong extrapolation expedites alignment. *arXiv preprint arXiv:2404.16792*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023.

Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*.

# A Appendix

## A.1 Data Details

| Subset | Training set | | | | Test |
|---|---|---|---|---|---|
| | $\mathcal{D}^2$ | $\mathcal{D}^3$ | $\mathcal{D}_Q^2$ | $\mathcal{D}_Q^3$ | |
| $\text{Harmless}_{\text{base}}$ | 42,536 | | 2,127 | | 2,312 |
| $\text{Helpful}_{\text{base}}$ | 43,835 | | 2,192 | | 2,354 |
| $\text{Helpful}_{\text{online}}$ | 22,002 | | 1,101 | | 1,137 |
| $\text{Helpful}_{\text{rejection}}$ | 52,420 | | 2,621 | | 2,749 |
| Total | 160,793 | | 8,041 | | 8,552 |

## A.2 DEFT Experiment Details

| Parameter | DEFT-PRO | DEFT-DPO |
|---|---|---|
| Epoch | 2 | 2 |
| SFT weight | 5e-2 | 5e-2 |
| Learning rate | 5e-6 | 5e-7 |
| Input length | 512 | 512 |
| Inference length | 128 | 128 |
| $\omega$ | 1.2e-6 | 1.2e-7 |
| $\beta$ | - | 0.1 |