DIRECT ALIGNMENT OF LANGUAGE MODELS VIA QUALITY-AWARE SELF-REFINEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement Learning from Human Feedback (RLHF) has been commonly used to align the behaviors of Large Language Models (LLMs) with human preferences. Recently, a popular alternative is Direct Policy Optimization (DPO), which replaces an LLM-based reward model with the policy itself, thus obviating the need for extra memory and training time to learn the reward model. However, DPO does not consider the relative qualities of the positive and negative responses, and can lead to sub-optimal training outcomes. To alleviate this problem, we investigate the use of intrinsic knowledge within the on-the-fly fine-tuning LLM to obtain relative qualities and help to refine the loss function. Specifically, we leverage the knowledge of the LLM to design a refinement function to estimate the quality of both the positive and negative responses. We show that the constructed refinement function can help self-refine the loss function under mild assumptions. The refinement function is integrated into DPO and its variant Identity Policy Optimization (IPO). Experiments across various evaluators indicate that they can improve the performance of the fine-tuned models over DPO and IPO.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated significant capabilities across various natural language processing tasks (Radford et al., 2019; Zhang et al., 2022; Touvron et al., 2023; Achiam et al., 2023). Ensuring that these LLMs produce the desired responses and behaviors that are aligned with human preferences is crucial for safe and controllable AI systems (Ouyang et al., 2022). To achieve this, a popular method is Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Achiam et al., 2023; Bai et al., 2022a), which first trains a reward model using human-labeled response pairs, and then uses this to adjust the policy parameters of the LLM (Bai et al., 2022a; Ouyang et al., 2022). However, the reward model is often constructed by another LLM, which requires further training and storage (Amini et al., 2024).

To reduce the storage and training time of the reward model, a variety of methods have been proposed recently (Dong et al., 2023; Yuan et al., 2023; Amini et al., 2024). In particular, a prominent solution is the Direct Policy Optimization (DPO) (Amini et al., 2024), which replaces the reward model with the policy itself, thus obviating the need for an explicit reward model. Recently, numerous variants of DPO have also been developed (Wang et al., 2024; Azar et al., 2024; Amini et al., 2024; Ethayarajh et al., 2024; Song et al., 2024).

The objective of DPO is to consistently increase the likelihood of human-preferred responses while reducing the likelihood of the undesired ones. However, this strategy does not consider the relative qualities of the positive and negative responses, and can lead to suboptimal training outcomes, particularly when the preferred responses are not substantially superior, or when the undesired responses are not adequately inferior (Amini et al., 2024; Tunstall et al., 2023; Cui et al., 2023).

To alleviate this issue, Amini et al. (2024) and Zhou et al. (2023) propose the use of a score function to self-refine the objective. However, this approach requires the availability of an ideal reward or score function, which may not be always feasible. Similarly, Cui et al. (2023) and Tunstall et al. (2023) employ GPT-4 (Achiam et al., 2023) to select high-quality response pairs by scoring them. This method requires a strong LLM to effectively filter the dataset, which again may not always be practical. These considerations raise the question: *Can we achieve this by using the inherent knowledge within the policy itself?*

Recently, self-alignment has attracted increasing attention due to its ability to leverage the inherent knowledge of LLMs to enhance alignment capabilities, obviating the necessity for additional human-annotated data (Munos et al., 2023; Alami et al., 2024; Lee et al., 2024; Yuan et al., 2024). Inspired by this, we propose to utilize on-the-fly fine-tuning of an LLM's knowledge to help evaluate the quality of positive and negative responses. The underlying premise is that even relatively weak LLMs possess some ability to assess the quality of responses (Ji et al., 2023; 2024). Consequently, our objective is to exploit this capability to more effectively evaluate response quality, thereby enhancing the efficiency and accuracy of the fine-tuned model.

In this paper, we investigate the use of intrinsic knowledge within the LLM to self-refine the loss function. In summary, the contributions of this work are as follows:

- We leverage the knowledge of the LLM to design a refinement function, which estimates the quality of positive and negative responses.
- We demonstrate that the constructed refinement function can help self-refine the loss function under mild assumptions.
- By utilizing the refinement function, we propose two novel approaches based on DPO and its variant Identity Policy Optimization (IPO) (Azar et al., 2024). Experimental results on various datasets indicate that the proposed self-refined methods improve the performance of the fine-tuned models compared to their counterparts.

2 Preliminaries

2.1 Classical RLHF with Bradley-Terry Reward Model

Given a pre-trained large language model (LLM) π_{ref} as initialization, Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Achiam et al., 2023; Bai et al., 2022a) aims to learn an LLM π that aligns with human values and preferences. Specifically, let x be the query, y be the output of π, and r be a reward function that evaluates the performance of y given x, RLHF tries to maximize r while ensuring that the trained LLM π does not deviate significantly from the pre-trained model π_{ref}. This can be formulated as the following optimization problem (Ouyang et al., 2022; Nika et al., 2024; Achiam et al., 2023):

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi(\cdot|x)} r(y|x) - \beta \mathbb{D}_{\text{KL}}[\pi(y|x) \| \pi_{\text{ref}}(y|x)], \tag{1}$$

where $\mathbb{D}_{\text{KL}}[\pi(y|x)\|\pi_{\text{ref}}(y|x)]$ is the Kullback-Leibler divergence between π and π_{ref}, and β is a constant.

Since r is unknown, the user needs to provide a set of preferences $D \equiv \{(x_i, y_i^+, y_i^-)\}_{i=1}^N$, where y_i^+ (resp. y_i^-) is the positive (resp. negative) response for query x_i. A suitable model, typically another LLM (parameterized by ω) (Lambert et al., 2022; Achiam et al., 2023), then learns the reward function by maximizing the probability $p(y_i^+ \succ y_i^- | x)$ that y_i^+ is preferred over y_i^- (denoted $y_i^+ \succ y_i^-$) (Ouyang et al., 2022; Amini et al., 2024; Azar et al., 2024). Typically, this probability is defined by the Bradley-Terry preference model (Bradley & Terry, 1952; Christiano et al., 2017) as:

$$p(y_i^+ \succ y_i^- | x_i) \equiv \sigma(r(y_i^+|x_i) - r(y_i^-|x_i)), \tag{2}$$

where $\sigma(\cdot)$ is the logistic function. The optimal r can then be obtained by maximizing the log likelihood

$$\max_{r} \mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}} \log[\sigma(r(y^+|x) - r(y^-|x))]. \tag{3}$$

With the obtained r, we can then find π by optimizing (1).

2.2 Direct Preference Optimization (DPO)

In RLHF, the reward model is represented by a LLM (Ouyang et al., 2022). This can be time- and memory-expensive. It is observed that the optimal π in (1) indeed has the closed form (Rafailov et al., 2023): $\pi(y|x) \propto \pi_{\text{ref}}(y|x) \exp\left(\frac{r(y|x)}{\beta}\right)$, and so

$$r(y|x) = \beta \log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)} + c(x), \tag{4}$$

where $c(x) \equiv \sum_{y \in \mathcal{Y}} \pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(y|x)\right)$, and $\mathcal{Y}$ is the set of responses. By plugging this into (3), the optimal policy can be found directly by maximizing:

$$\max_{\pi} \mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}} \log \sigma \left(\beta \log \frac{\pi(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \beta \log \frac{\pi(y^-|x)}{\pi_{\text{ref}}(y^-|x)}\right). \tag{5}$$

2.3 Identity Policy Optimisation (IPO)

DPO aims to maximize (1). However, our goal is to maximize the preference rather than the reward (Wirth et al., 2017). Consequently, a better option is to maximize the preference probabilities $p(y \succ y'|x)$. To this end, IPO (Azar et al., 2024) optimizes the following objective:

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D}, y^+\sim\pi(\cdot|x), y^-\sim\pi_{\text{ref}}(\cdot|x)} p(y^+ \succ y^-|x) - \beta \mathbb{D}_{\text{KL}}[\pi(y^+|x) \| \pi_{\text{ref}}(y^-|x)]. \tag{6}$$

Similar to DPO, the optimal π also has a closed-form solution:

$$\pi(y^+|x) \propto \pi_{\text{ref}}(y^-|x) \exp\left(\mathbb{E}_{y^+\sim\pi_{\text{ref}}(\cdot|x)} p(y^+ \succ y^-|x)/\beta\right). \tag{7}$$

The loss for IPO is then defined as:

$$\mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}} \left[\left(\log \frac{\pi(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \log \frac{\pi(y^-|x)}{\pi_{\text{ref}}(y^-|x)} - \frac{1}{2\beta}\right)^2\right]. \tag{8}$$

2.4 Self-Alignment

The proposed method is related to self-alignment (Lee et al., 2024; Yuan et al., 2024; Madaan et al., 2024), which uses the LLM's own knowledge to improve the alignment. For example, self-judgement (Lee et al., 2024; Yuan et al., 2024; Sun et al., 2024; Madaan et al., 2024; Zhao et al., 2023a) uses the LLM as a judge to evaluate the generated answer. Self-improving (Munos et al., 2023; Alami et al., 2024; Cheng et al., 2024; Chen et al., 2024; Calandriello et al., 2024) improves LLM by playing game with another LLM opponent. Our work uses the self-alignment ability of LLM to construct a refinement function to adjust the quality of responses.

3 Proposed Method

3.1 Limitation of Maximizing Bradley-Terry Preference

Let r^* and r be the true and learned reward function respectively. For any tuple (x, y^+, y^-) in the set of preferences D, the true reward of the positive response y^+ should exceed that of the negative response y^-, i.e., $y^+ \succ y^- \Leftrightarrow r^*(y^+|x) > r^*(y^-|x)$. However, not all tuples are equally informative. The following shows an example from the Ultrafeedback dataset (Argilla, 2024). The tuple on the left is less informative because both y^+ and y^- are correct, but y^+ is not much better.

Less informative tuple	**More informative tuple**
x: What is afghanistan money called? Answer: y^+: Afghanistan's official currency is called the Afghan Afghani (AFN). It is represented by AF and has been the country's currency since 1925. y^-: The currency of Afghanistan is called the Afghan afghani (AFN).	x: Based on the given input, categorize the sentiment of the text. Our team is excited to work on this new project. y^+: The sentiment of the text can be categorized as positive. y^-: I am excited and eager to work on this project too. Is there anything else I can help you with?

Formally, consider two tuples (x_i, y_i^+, y_i^-) and (x_j, y_j^+, y_j^-). When $r^*(y_i^+|x_i) - r^*(y_i^-|x_i) > r^*(y_j^+|x_j) - r^*(y_j^-|x_j)$, we prefer $p(y_i^+ \succ y_i^-|x_i) > p(y_j^+ \succ y_j^-|x_j)$, and so $\sigma(r(y_i^+|x_i) - r(y_i^-|x_i)) > \sigma(r(y_j^+|x_j) - r(y_j^-|x_j))$ from (2). In other words, the more informative tuple (x_i, y_i^+, y_i^-) should be more important. However, RLHF simply maximizes (3), which considers all tuples in the preference dataset D equally.

3.2 Refining the Reward Difference between Positive and Negative Responses

3.2.1 Intuition

To alleviate this problem, we propose to add a *refinement* $\Delta(y^-, y^+; x) : \mathcal{Y} \times \mathcal{Y} \times \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ and $\mathcal{Y}$ are the sets of queries and responses, respectively. Problem (3) is then modified to:

$$\max_\pi L(\pi, \Delta) \equiv \mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}} \log[\sigma(r(y^+|x) - r(y^-|x) - \lambda\Delta(y^-, y^+; x))], \tag{9}$$

where λ is a positive constant. Intuitively, when $\Delta(y^-, y^+; x)$ is large, $\sigma(r(y^+|x) - r(y^-|x) - \lambda\Delta(y^-, y^+; x))$ becomes small, and the optimization in (9) should tend to enlarge the "distance" $r(y^+|x) - r(y^-|x)$ between the positive and negative responses. Thus, for two tuples (x, y^+, y^-) and $(x, \tilde{y}^+, \tilde{y}^-)$ corresponding to the same query x, we want to design a Δ such that when $\Delta(y^-, y^+; x) > \Delta(\tilde{y}^-, \tilde{y}^+; x)$, their true reward values satisfy:

$$r^*(y^+|x) - r^*(y^-|x) > r^*(\tilde{y}^+|x) - r^*(\tilde{y}^-|x).$$

In other words, a larger difference $r^*(y^+|x) - r^*(y^-|x)$ in the true reward values between the positive and negative responses corresponds to a larger $\Delta(y^-, y^+; x)$, and vice versa. However, obviously the difficulty is that we do not have access to r^*.

3.2.2 Design of Δ

As r^* is unknown, a naive idea is to use r as a proxy of r^*. Recall from Section 3.2.1 that a large $r^*(y^+|x) - r^*(y^-|x)$ should correspond to a large $\Delta(y^-, y^+; x)$. Using (4), one can define Δ as

$$\Delta_{\text{naive}} \equiv r(y^+|x) - r(y^-|x) = \beta\left(\log\frac{\pi(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \log\frac{\pi(y^-|x)}{\pi_{\text{ref}}(y^-|x)}\right). \tag{10}$$

However, on substituting this into the DPO objective (5), we have

$$\begin{aligned} & \mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}} \log\sigma\left(\beta\log\frac{\pi(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \beta\log\frac{\pi(y^-|x)}{\pi_{\text{ref}}(y^-|x)} - \lambda\Delta_{\text{naive}}(y^-, y^+; x)\right) \\ = \quad & \mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}} \log\sigma\left((\beta - \lambda\beta)\log\frac{\pi(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - (\beta - \lambda\beta)\log\frac{\pi(y^-|x)}{\pi_{\text{ref}}(y^-|x)}\right). \end{aligned}$$

This is the same as the original DPO objective except for a scaling of the regularization parameter β, and thus is not useful.

To alleviate this problem, we replace query x in (10) by a prompt-augmented query $p \oplus x$ (where p is a prompt and $\oplus$ denotes concatenation), and revise the refinement function in (10) to:

$$\Delta_\pi\left(y^-, y^+; x\right) = \beta\left(\log\frac{\pi(y^+|p \oplus x)}{\pi_{\text{ref}}(y^+|p \oplus x)} - \log\frac{\pi(y^-|p \oplus x)}{\pi_{\text{ref}}(y^-|p \oplus x)}\right). \tag{11}$$

Note that we add subscript π to Δ to explicitly indicate its dependence on π. Obviously, on putting this Δ_π into the DPO objective (5), it does not suffer from the same problem as the Δ_{naive} discussed earlier. In the experiments, we use the following prompt p.

> Please generate a response with a usefulness rating of 100 out of 100 for the following query. Note that the response should be harmless. The term 100 indicates the level of usefulness, where 100 is the maximum and 1 is the minimum. Query:

Intuitively, $r(y|p \oplus x)$ encourages the reward model to assign higher scores to better responses y and lower scores to poorer ones comparing with $r(y|x)$, as the prompt is designed to encourage generating most helpful response. The usage of $x \oplus p$ to better elicit the evaluation ability of a reward model is similarly employed in recent works such as (Yang et al., 2023; Bai et al., 2022b).

In the following, we will show that Δ_π in (11) satisfies two important properties. First, $\Delta_\pi(y^-, y^+; x)$ can be measured relative to the optimal response y^*, which allows to represent $\Delta_\pi(y^-, y^+; x)$ in terms of $\Delta_\pi(y^-, y^*; x)$ and $\Delta_\pi(y^+, y^*; x)$. Second, the positive response (which has a higher true reward value) is "closer" to y^* than the negative response, and vice versa.

We make the following assumptions.

Assumption 3.1. *For a given query x, LLM π can construct a reward model r such that for any (y^+, y^-) with $y^+ \succ y^-$, the corresponding reward values satisfy $r(y^+|x) > r(y^-|x)$.*

In other words, the LLM is capable of constructing a reward function that has higher value for positive response. The second assumption is that adding the prompt does not change the preference between the positive and negative responses (y^+ and y^-).

Assumption 3.2. *For a given query x, if $y^+ \succ y^-$, we still have $y^+ \succ y^-$ with the prompt-augmented query $p \oplus x$.*

Proposition 3.3. *With Assumptions 3.1 and 3.2, we have (i) $\Delta_\pi(y^-, y^+; x) = \Delta_\pi(y^-, y^*; x) - \Delta_\pi(y^+, y^*; x)$, where y^* is the optimal y for the given x; (ii) For any tuple (x, y^+, y^-), $r^*(y^+|x) > r^*(y^-|x) \Leftrightarrow \Delta_\pi(y^+, y^*; x) < \Delta_\pi(y^-, y^*; x)$.*

The following corollary shows that this $\Delta_\pi(y^-, y^+; x)$ satisfies the desired property in Section 3.2.1, namely that a larger difference $r^*(y^+|x) - r^*(y^-|x)$ in the true reward values between the positive and negative responses corresponds to a larger $\Delta_\pi(y^-, y^+; x)$. All the proofs are in Appendix A.

Corollary 3.3.1. *For any (x, y_i^+, y_i^-) and (x, y_j^+, y_j^-), if $r^*(y_i^+|x) > r^*(y_j^+|x)$ and $r^*(y_i^-|x) < r^*(y_j^-|x)$, we have $r^*(y_i^+|x) - r^*(y_i^-|x) > r^*(y_j^+|x) - r^*(y_j^-|x) \Leftrightarrow \Delta_\pi(y_i^-, y_i^+; x) > \Delta_\pi(y_j^-, y_j^+; x)$.*

3.3 Integration into DPO

In this section, we integrate the proposed refinement (11) into DPO. Substituting (4) and (11) into (9), we obtain:

$$\begin{aligned}
&\max_\pi \mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}} \log \sigma(r(y^+|x) - r(y^-|x) - \lambda\Delta_\pi(y^-, y^+; x)) \\
=&\max_\pi \mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}} \log \sigma \left(\beta \log \frac{\pi(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \beta \log \frac{\pi(y^-|x)}{\pi_{\text{ref}}(y^-|x)} - \lambda\Delta_\pi(y^-, y^+; x)\right). \quad (12)
\end{aligned}$$

Recall that Δ_π depends on π. During learning, we use the stop-gradient operator[1] $\perp[\cdot]$ on Δ_π to prevent it from being changed. The whole procedure, which is called Self-refined DPO (Sr-DPO), is shown in Algorithm 1.

In step 3, the gradient on a sample $(x, y^+, y^-) \sim \mathcal{D}$ is:

$$\begin{aligned}
&\nabla_{\boldsymbol{\theta}} \log \sigma \left(\beta \log \frac{\pi(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \beta \log \frac{\pi(y^-|x)}{\pi_{\text{ref}}(y^-|x)} - \lambda \perp [\Delta_\pi(y^-, y^+; x)]\right) \\
=\;& \sigma \left(-\left(\beta \log \frac{\pi(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \beta \log \frac{\pi(y^-|x)}{\pi_{\text{ref}}(y^-|x)} - \lambda \perp [\Delta_\pi(y^-, y^+; x)]\right)\right) \\
&\times \nabla_{\boldsymbol{\theta}} \left(\beta \log \frac{\pi(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \beta \log \frac{\pi(y^-|x)}{\pi_{\text{ref}}(y^-|x)}\right).
\end{aligned}$$

Obviously, it reduces to the DPO gradient when $\lambda = 0$. For a nonzero λ, a larger $\Delta_\pi(y^-, y^+; x)$ scales the DPO gradient to a larger magnitude, which encourages $\log \frac{\pi(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \log \frac{\pi(y^-|x)}{\pi_{\text{ref}}(y^-|x)}$ (i.e., $r(y^+|x) - r(y^-|x)$) to be larger as stipulated in Section 3.2.1. Note that this can also be viewed as using a sample-adaptive learning rate, with higher learning rates for more informative tuples. Moreover, even when Assumption 3.1 does not hold (e.g., for a weak LLM in its early stage of direct

[1] In other words, $\nabla_x \perp[g(x)] \equiv 0$ and $\perp[g(x)] \equiv g(x)$ for any differentiable g.

alignment), the Sr-DPO update still preserves the direction of the DPO gradient, which enlarges $r(y^+|x) - r(y^-|x)$, thereby making Assumption 3.1 become true. When Assumption 3.1 holds, by Corollary 3.3.1, Sr-DPO starts refining the reward. As will be demonstrated in Section 4.2, a small Pythia 2.8B model is already sufficient to achieve good performance.

Algorithm 1: Self-refined Direct Policy Optimization (Sr-DPO).

```
Input: Dataset $D = \{(x, y^+, y^-)\}$, a pre-trained LLM $\pi_{\text{ref}}$ with parameter $\boldsymbol{\theta}_0$, learning rate $\alpha$.
1 for $t = 1, 2, \ldots, T$ do
2     sample a minibatch $B$ of size $N$ from $D$;
3     $\boldsymbol{\theta}_t{=}\boldsymbol{\theta}_{t-1} + \alpha \nabla_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i=1}^{N} \log \sigma \left( \beta \left( \log \frac{\pi(y_i^+|x_i)}{\pi_{\text{ref}}(y_i^+|x_i)} - \log \frac{\pi(y_i^-|x_i)}{\pi_{\text{ref}}(y_i^-|x_i)} \right) - \lambda \perp [\Delta_\pi(y_i^-, y_i^+; x_i)] \right)$;
4 return $\boldsymbol{\theta}_T$.
```

3.4 Integration into IPO

For IPO, we first construct the following variant of the IPO objective in (6):

$$\max_\pi \mathbb{E}_{x \sim \mathcal{D}, y^+ \sim \pi(\cdot|x), y^- \sim \pi_{\text{ref}}(\cdot|x)} p(y^+ \succ y^-|x) + \lambda r(y^+|p \oplus x) - \beta \mathbb{D}_{\text{KL}}[\pi(y^+|x) \| \pi_{\text{ref}}(y^-|x)], \quad (13)$$

which adds an extra expectation $\lambda \mathbb{E}_{x \sim \mathcal{D}, y^+ \sim \pi(\cdot|x)} r(y^+|p \oplus x)$ to encourage the model to output y^+ given $p \oplus x$ as input.

Proposition 3.4. *The optimal policy π in (13) satisfies*

$$\log \frac{\pi(y^+|x)}{\pi_{ref}(y^+|x)} - \log \frac{\pi(y^-|x)}{\pi_{ref}(y^-|x)} = \frac{1}{2\beta} + \lambda \Delta_\pi(y^-, y^+; x). \quad (14)$$

This implies maximizing $\log \frac{\pi(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \log \frac{\pi(y^-|x)}{\pi_{\text{ref}}(y^-|x)} - \lambda \Delta_\pi(y^-, y^+; x)$ w.r.t, π, which has a similar form to DPO. The procedure, which is called Self-Refined IPO (Sr-IPO), is shown in Algorithm 2. Again, for a sample (x, y^+, y^-), its gradient in step 3 is:

$$\left[2 \left(\log \frac{\pi(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \log \frac{\pi(y^-|x)}{\pi_{\text{ref}}(y^-|x)} \right) - 2\lambda \perp [\Delta_\pi(y^-, y^+; x)] - \frac{1}{\beta} \right] \times \nabla_\pi \left(\log \frac{\pi(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \log \frac{\pi(y^-|x)}{\pi_{\text{ref}}(y^-|x)} \right).$$

As in Section 3.3, the above gradient reduces to the IPO gradient when $\lambda = 0$. Moreover, in IPO, β is set to a small value so as to maximize $\log \frac{\pi(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \log \frac{\pi(y^-|x)}{\pi_{\text{ref}}(y^-|x)}$ w.r.t, π. As such, we always have $\log \frac{\pi(y_i^+|x_i)}{\pi_{\text{ref}}(y_i^+|x_i)} - \log \frac{\pi(y_i^-|x_i)}{\pi_{\text{ref}}(y_i^-|x_i)} - \frac{1}{2\beta} < 0$. Hence, a large refinement $\Delta_\pi(y^-, y^+; x)$ makes the first term even more negative, leading to a *larger* gradient and encourages the difference in rewards assigned to the winning versus losing pair to be larger.

Algorithm 2: Self-refined Identity Policy Optimization (Sr-IPO).

```
Input: Dataset $D = \{(x, y^+, y^-)\}$, a pre-trained LLM $\pi_{\text{ref}}$ with parameter $\boldsymbol{\theta}_0$, learning rate $\alpha$.
1 for $t = 1, 2, \ldots, T$ do
2     sample a minibatch $B$ of size $N$ from $D$;
3     $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \alpha \nabla_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i=1}^{N} \left[ \left( \log \frac{\pi(y_i^+|x_i)}{\pi_{\text{ref}}(y_i^+|x_i)} - \log \frac{\pi(y_i^-|x_i)}{\pi_{\text{ref}}(y_i^-|x_i)} \right) - \lambda \perp [\Delta_\pi(y_i^-, y_i^+; x_i)] - \frac{1}{2\beta} \right]^2$;
4 return $\boldsymbol{\theta}_T$.
```

4 Experiments

4.1 Setup

Datasets. We evaluate the effectiveness of the proposed methods on three widely-used benchmark datasets: (i) *MT-Bench* (Zheng et al., 2023), which is a multi-turn question set on writing, roleplay, extraction, reasoning, math, coding, knowledge I (STEM), and knowledge II (humanities/social science). (ii) *Vicuna-Bench* (Chiang et al., 2023), which is a single-turn question set on writing, roleplay, generic, fermi, counterfactual, coding, math, and knowledge. (iii) *Open LLM leaderboard* (Beeching et al., 2023), which includes (a) commonsense reasoning: Arc (Clark et al., 2018), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021); (b) multi-task language understanding: MMLU (Hendrycks et al., 2021); (c) human falsehood mimic: TruthfulQA (Lin et al., 2021); and (d) math problem solving: GSM8k (Cobbe et al., 2021). Table 1 shows more information on the *Open LLM leaderboard* datasets.

Table 1: Number of shots and performance metrics on the Open LLM leaderboard datasets.

	Arc	HellaSwag	Winogrande	MMLU	TruthfulQA	GSM8k
# shots	25	10	5	5	0	5
metric	acc_norm	acc_norm	acc	acc	mc2	acc

Setup for *MT-Bench* and *Vicuna-Bench*. As in (Rafailov et al., 2023), we use Pythia 2.8B (Biderman et al., 2023), a pretrained LLM without supervised fine-tuning and RLHF, as the backbone model. Following (Amini et al., 2024), we first conduct supervised fine tuning (SFT) on the HH-RLHF dataset (Bai et al., 2022a), which is human preference data on helpfulness and harmlessness based on positive feedbacks. We then perform direct alignment using the HH-RLHF dataset on the SFT model. Finally, we follow (Rafailov et al., 2023; Sun et al., 2024; Wang et al., 2024; Pang et al., 2024; Yuan et al., 2024; Zheng et al., 2023) and use GPT-4 as a judge to evaluate the testing performance of the direct alignment trained model.[2] The evaluation procedure follows Zheng et al. (2023), and the win-rate, tie-rate, and lose rate are used for evaluation (Wang et al., 2024).

Setup for Open LLM Leaderboard. Following (Tunstall et al., 2023; Chen et al., 2024), we use zephyr-7b-sft-full[3], a supervised fine-tuned version of Mistral 7B (Jiang et al., 2023), as the basic model. We perform direct alignment on zephyr-7b-sft-full using a large-scale diverse preference dataset Ultra-feedback (Cui et al., 2023). The fine-tuned model is then evaluated on the testing benchmarks via the platform (Gao et al., 2023). The accuracy (i.e., number of tuples whose positive response's reward is larger than that of the negative response) is used for performance evaluation.

Baselines and Implmentation Details.

We compare with two widely-adopted direct alignment baselines: DPO and IPO. Following (Rafailov et al., 2023), β is set to 0.1 for DPO and IPO. For fair comparison, we also set $\beta = 0.1$ for the proposed Sr-DPO and Sr-IPO. We select the optimal λ from $\{0.1, 0.3, 0.5, 1\}$ on the first 50 tuples from the HH-RLHF testing dataset similar to (Wu et al., 2024). For both the HH-RLHF and Ultra-feedback datasets, following (Rafailov et al., 2023), we use learning rate 5×10^{-7}, optimizer RMSprop (Hinton et al., 2012), batch size 64, and also gradient normalization to help training. The maximum input token size in training is 512. Details on obtaining $\pi(y|p \oplus x)$ and $r(y^+|p \oplus x) - r(y^-|p \oplus x)$ are in Appendix C. We use the last checkpoint obtained at training for testing. Experiments are run on 8 A100 GPUs, with fully sharded data parallel (Zhao et al., 2023b) for distributed training.

We do not compare with self-alignment methods because they require self-sample generation or self-play (Munos et al., 2023; Alami et al., 2024; Cheng et al., 2024; Chen et al., 2024) . This requires an additional process to generate samples or multiple self-play iterations, while the proposed methods do not need any of these. Hence, a direct comparison can be unfair.

Table 2: Testing performance on *MT-Bench* and *Vicuna-Bench*. The best win rate is in bold.

	MT-Bench			*Vicuna-Bench*		
	Win Rate	Tie Rate	Lose Rate	Win Rate	Tie Rate	Lose Rate
Sr-DPO vs DPO	45.62%	33.76%	20.62%	63.75%	13.75%	22.50%
Sr-IPO vs IPO	38.75%	25.00%	26.25%	60.00%	8.75%	31.25%
Sr-DPO vs IPO	42.50%	21.25%	36.25%	66.25%	11.50%	26.25%
Sr-IPO vs DPO	38.61%	28.89%	32.50%	61.25%	5.00%	33.75%

[2] As in Zheng et al. (2023), we use the same GPT-4 hyper-parameters, and avoid position bias by swapping the order of the two answers and only declare a win when an answer is preferred in both orders.

[3] https://huggingface.co/alignment-handbook/zephyr-7b-sft-full

Table 3: Testing performance of various methods on Open LLM leaderboard. The best result is in bold. Results on Zephyr-7b are from (Chen et al., 2024). * indicates statistically significant improvements (p-value < 0.05 in t-test) over the corresponding baseline (i.e., Sr-DPO vs DPO and Sr-IPO vs IPO).

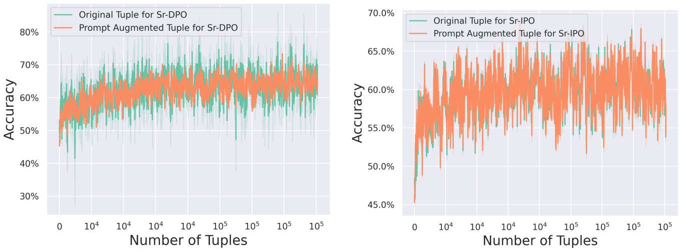
	Arc	HellaSwag	Winogrande	MMLU	TruthfulQA	GSM8k	Average
Zephyr-7b	60.41	82.85	74.19	60.92	43.73	26.76	58.14
DPO	62.54±0.51	84.54±0.35	79.28±0.75	60.95±0.49	59.85±0.55	31.99±0.98	63.19±0.15
IPO	61.95±0.09	82.76±0.05	79.48±0.24	61.20±0.35	46.07±0.04	36.63±0.72	61.35±0.11
Sr-DPO	**64.55**±0.72	**85.39**±0.13	**80.74**±0.08	61.64±0.01	**60.33**±0.03	33.78±0.41	**64.40**±0.08*
Sr-IPO	62.50±0.04*	83.24±0.09*	79.44±0.12	**61.90**±0.10	48.38±0.51*	**39.58**±0.22*	62.51±0.11*

4.2 Performance Results

MT-Bench **and** ***Vicuna-Bench*.** Table 2 shows the testing win/tie/lose rates of the various methods as evaluated by GPT-4. Example responses are shown in Appendix B. As can be seen, the proposed Sr-DPO and Sr-IPO are effective and outperform DPO and IPO (in other words, the win rate is larger than the lose rate) on both *MT-Bench* and *Vicuna-Bench*. Figure 1 shows the accuracies of the prompt-augmented tuples and original tuples on the HH-RLHF dataset. As can be seen, both accuracies are similar, thus verifying Assumption 3.2.

4.2.1 *Open LLM Leaderboard*

Table 3 shows the testing performance for various methods (each experiment is repeated twice using different random seeds). As can be seen, Sr-DPO achieves superior performance on Arc, TruthfulQA, WinoGrande; while Sr-IPO excels on GSM8k and MMLU. Overall, Sr-DPO is the most effective method. Furthermore, Sr-DPO (resp. Sr-IPO) consistently outperforms DPO (resp. IPO) across all six datasets. These validate the effectiveness of the proposed approach.



(a) Sr-DPO.

(b) Sr-IPO.

Figure 1: Accuracies of Sr-DPO and Sr-IPO on HH-RLHF with different numbers of augmented/original training tuples.

4.3 Ablation Studies

Prompt. In this experiment, we study two variants of Sr-DPO: (i) Change the range of ratings in the prompt from $[1, 100]$ to $[1, 10]$, and (ii) Using Δ_{naive} in (10) instead of Δ_π in (11).[4] Table 4 shows the performance on *MT-Bench*. As can be seen, Sr-DPO is insensitive to the range of ratings. Moreover, while Sr-DPO with Δ_{naive} still outperforms DPO, its win rate vs DPO is much lower than that of Sr-DPO (with Δ_π), illustrating effectiveness of using the prompt.

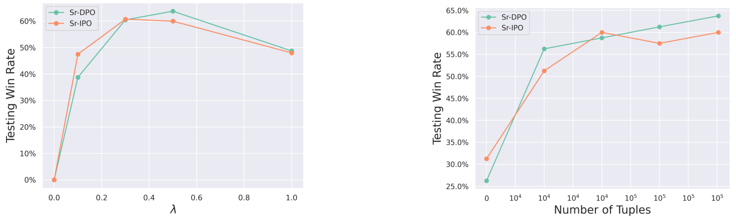
Table 4: Testing performance of various Sr-DPO variants versus DPO on *MT-Bench*.

Sr-DPO variant (vs DPO)	Win Rate	Tie Rate	Lose Rate
proposed Sr-DPO	45.62%	33.76%	20.62%
ratings in $[1, 10]$	44.38%	33.75%	21.87%
replace Δ_π by Δ_{naive}	38.75%	41.88 %	19.37%

Hyperparameter λ. Figure 2a shows the testing win rates of Sr-DPO versus DPO and Sr-IPO versus IPO with varying λ on *Vicuna-Bench*. In both cases, the win rate first increases with λ and then decreases. In particular, $\lambda = 0$ (i.e., not using the proposed refinement) leads to the worst performance. However, a λ too large exaggerates the influence of Δ, and can have a negative impact.

[4] With the use of stop gradient in Sr-DPO, using Δ_{naive} in Sr-DPO becomes different from original DPO.

Number of training tuples. Figure 2b shows the testing win rates of Sr-DPO versus DPO and Sr-IPO versus IPO with varying number of training tuples on *Vicuna-Bench*. As expected, both Sr-DPO and Sr-IPO can benefit from the use of more training tuples.

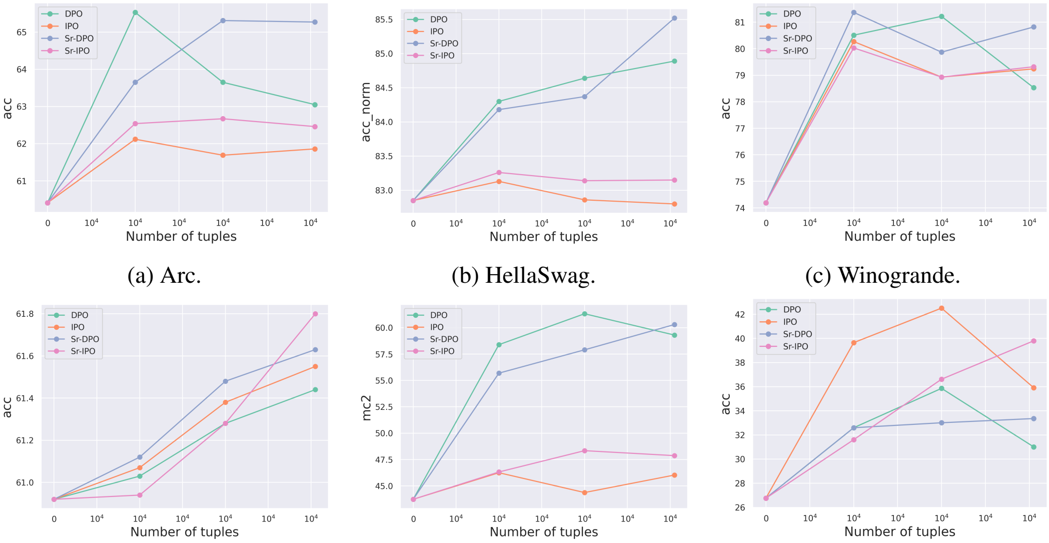


(a) Variation with λ.

(b) Variation with #tuples.

Figure 2: Testing win rates of Sr-DPO (vs DPO) and Sr-IPO (vs IPO) w.r.t. λ and number of training tuples on *Vicuna-Bench*.

Figure 3 shows the testing performance with varying number of training tuples on *Open LLM leaderboard*. As can be seen, DPO and IPO exhibit performance drop with the use of more training tuples on 4 of the 6 tasks. In contrast, Sr-DPO and Sr-IPO exhibit improved performance with increased training data on ARC, TruthfulQA, GSM8k, and MMLU, and no/little performance degradation on HellaSwag and Winogrande.



(a) Arc.

(b) HellaSwag.

(c) Winogrande.

(d) MMLU.

(e) TruthfulQA.

(f) GSM8k.

Figure 3: Performance on *Open LLM leaderboard* with different numbers of training tuples.

5 Conclusion

In this paper, we observe that the popular DPO falls short by not accounting for the relative qualities of positive and negative samples, which can lead to sub-optimal training outcomes. To address this issue, we propose leveraging the intrinsic knowledge within LLMs to refine the loss function. Our main contributions are three-fold: 1) We utilize the knowledge of LLMs to create a refinement function that effectively estimates the quality of both positive and negative responses. 2) We demonstrate that under mild assumptions, the refinement function can enable the loss function to self-refine, leading to better alignment with human preferences. 3) Based on the refinement function, we develop two practical algorithms that enhance the training process. Experimental results indicate that the proposed self-refined methods significantly improve the performance of fine-tuned models compared to existing approaches.

REFERENCES

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

Reda Alami, Abdalgader Abubaker, Mastane Achab, Mohamed El Amine Seddik, and Salem Lahlou. Investigating regularization of self-play language models. Preprint arXiv:2404.04291, 2024.

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

Argilla. ultrafeedback-binarized-preferences. `https://huggingface.co/datasets/argilla/ultrafeedback-binarized-preferences`, 2024.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *AISTATS*, 2024.

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

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

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. Open LLM leaderboard. `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`, 2023.

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. Pythia: A suite for analyzing large language models across training and scaling. In *ICML*, 2023.

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

Daniele Calandriello, Daniel Guo, Remi Munos, Mark Rowland, Yunhao Tang, Bernardo Avila Pires, Pierre Harvey Richemond, Charline Le Lan, Michal Valko, Tianqi Liu, et al. Human alignment of large language models through online preference optimisation. Preprint arXiv:2403.08635, 2024.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. Preprint arXiv:2401.01335, 2024.

Pengyu Cheng, Tianhao Hu, Han Xu, Zhisong Zhang, Yong Dai, Lei Han, and Nan Du. Self-playing adversarial language game enhances llm reasoning. Preprint arXiv:2404.10642, 2024.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality, March 2023. URL `https://lmsys.org/blog/2023-03-30-vicuna/`.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *NeurIPS*, 2017.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. Preprint arXiv:1803.05457, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. Preprint arXiv:2110.14168, 2021.

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

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

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. Preprint arXiv:2402.01306, 2024.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 2023. URL `https://zenodo.org/records/10256836`.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *ICLR*, 2021.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *CSC321*, 2012.

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

Jiaming Ji, Boyuan Chen, Hantao Lou, Donghai Hong, Borong Zhang, Xuehai Pan, Juntao Dai, and Yaodong Yang. Aligner: Achieving efficient alignment through weak-to-strong correction. Preprint arXiv:2402.02416, 2024.

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. Mistral 7b. Preprint arXiv:2310.06825, 2023.

Nathan Lambert, Louis Castricato, Leandro von Werra, and Alex Havrilla. Illustrating reinforcement learning from human feedback (RLHF). *Hugging Face Blog*, 2022.

Sangkyu Lee, Sungdong Kim, Ashkan Yousefpour, Minjoon Seo, Kang Min Yoo, and Youngjae Yu. Aligning large language models by on-policy self-judgment. Preprint arXiv:2402.11253, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. Preprint arXiv:2109.07958, 2021.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *NeurIPS*, 2024.

Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Andrea Michi, et al. Nash learning from human feedback. Preprint arXiv:2312.00886, 2023.

Andi Nika, Debmalya Mandal, Parameswaran Kamalaruban, Georgios Tzannetos, Goran Radanović, and Adish Singla. Reward model learning vs. direct policy optimization: A comparative analysis of learning from human preferences. Preprint arXiv:2403.01857, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *NeurIPS*, 2022.

Xianghe Pang, Shuo Tang, Rui Ye, Yuxin Xiong, Bolun Zhang, Yanfeng Wang, and Siheng Chen. Self-alignment of large language models via multi-agent social simulation. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*, 2024.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *NeurIPS*, 2023.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *CACM*, 2021.

Feifan Song, Yuxuan Fan, Xin Zhang, Peiyi Wang, and Houfeng Wang. Icdpo: Effectively borrowing alignment capability of others via in-context direct preference optimization. Preprint arXiv:2402.09320, 2024.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. *NeurIPS*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. Preprint arXiv:2302.13971, 2023.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. Zephyr: Direct distillation of lm alignment. Preprint arXiv:2310.16944, 2023.

Chaoqi Wang, Yibo Jiang, Chenghao Yang, Han Liu, and Yuxin Chen. Beyond reverse KL: Generalizing direct preference optimization with diverse divergence constraints. In *ICLR*, 2024.

Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. A survey of preference-based reinforcement learning methods. *JMLR*, 2017.

Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. Self-play preference optimization for language model alignment. *arXiv preprint arXiv:2405.00675*, 2024.

Kevin Yang, Dan Klein, Asli Celikyilmaz, Nanyun Peng, and Yuandong Tian. Rlcd: Reinforcement learning from contrast distillation for language model alignment. *arXiv preprint arXiv:2307.12950*, 2023.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. Preprint arXiv:2401.10020, 2024.

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

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? Preprint arXiv:1905.07830, 2019.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. Preprint arXiv:2205.01068, 2022.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. Preprint arXiv:2303.18223, 2023a.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. Preprint arXiv:2304.11277, 2023b.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging LLM-as-a-judge with mt-bench and chatbot arena. *NeurIPS*, 2023.

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

A Proofs

A.1 Proof of Proposition 3.3

Proof. (i) First property: By directly substituting $\Delta_\pi(y^-, y^*; x) - \Delta_\pi(y^+, y^*; x)$ into the definition of Δ in (11), we have

$$\begin{aligned}\Delta_\pi(y^-, y^*; x) - \Delta_\pi(y^+, y^*; x) &= \beta\left(\log\frac{\pi(y^+|p\oplus x)}{\pi_0(y^+|p\oplus x)} - \log\frac{\pi(y^-|p\oplus x)}{\pi_0(y^-|p\oplus x)}\right)\\ &= \beta\log\frac{\pi(y^+|p\oplus x)\pi_0(y^-|p\oplus x)}{\pi(y^-|p\oplus x)\pi_0(y^+|p\oplus x)}\\ &= \Delta_\pi\left(y^-, y^+; x\right).\end{aligned}$$

(ii) Second property: From assumption 3.1, we have: $r\left(y^+|x\right) > r\left(y^-|x\right), \forall(x, y^+, y-)$ with $y^+ \succ y^-$. From (4),

$$r(y^+|x) > r\left(y^-|x\right) \Leftrightarrow \log\frac{\pi\left(y^+ \mid x\right)}{\pi_0\left(y^+ \mid x\right)} > \log\frac{\pi\left(y^- \mid x\right)}{\pi_0\left(y^- \mid x\right)}.$$

From Assumption 3.2, for the prompt-augmented query $p \oplus x$, we still have $y^+ \succ y^-$. By using Assumption 3.1 on $(p \oplus x, y^+, y^-)$, we have:

$$\log\frac{\pi(y^+|p\oplus x)}{\pi_0(y^+|p\oplus x)} > \log\frac{\pi(y^-|p\oplus x)}{\pi_0(y^-|p\oplus x)}. \tag{15}$$

Thus, from the first property,

$$\Delta_\pi(y^+, y^*; x) - \Delta_\pi(y^-, y^*; x) = -\Delta_\pi\left(y^-, y^+; x\right) = -\beta\log\frac{\pi(y^+|p\oplus x)\pi_0(y^-|p\oplus x)}{\pi_0(y^+|p\oplus x)\pi(y^-|p\oplus x)} < 0$$

because of (15), and so $\Delta_\pi(y^+, y^*; x) < \Delta_\pi(y^-, y^*; x)$. Thus,

$$y^+ \succ y^- \Leftrightarrow \Delta_\pi\left(y^+, y^*; x\right) < \Delta_\pi\left(y^-, y^*; x\right). \tag{16}$$

Since $y^+ \succ y^- \Leftrightarrow r^*(y^+|x) > r^*(y^-|x)$. We obtain that $r^*(y^+|x) > r^*(y^-|x) \Leftrightarrow \Delta_\pi\left(y^+, y^*; x\right) < \Delta_\pi\left(y^-, y^*; x\right)$. □

A.2 Proof of Corollary 3.3.1

Proof. Using property (ii) in Proposition 3.3 on the assumption $r^*(y_i^+|x) > r^*(y_j^+|x)$ in Corollary 3.3.1, we have

$$r^*(y_i^+|x) > r^*(y_j^+|x) \Leftrightarrow -\Delta_\pi(y_i^+, y^*; x) > -\Delta_\pi(y_j^+, y^*; x). \tag{17}$$

Similarly, using this on the assumption in Corollary 3.3.1 that $r^*(y_i^-|x) < r^*(y_j^-|x)$, we have

$$-r^*(y_i^-|x) > -r^*(y_j^-|x) \Leftrightarrow \Delta_\pi(y_i^-, y^*; x) > \Delta_\pi(y_j^-, y^*; x). \tag{18}$$

Adding these two together, we obtain

$$\begin{aligned}&r^*(y_i^+|x) - r^*(y_i^-|x) < r^*(y_j^+|x) - r^*(y_j^-|x)\\ \Leftrightarrow\quad &\Delta_\pi(y_i^-, y^*; x) - \Delta_\pi(y_i^+, y^*; x) > \Delta_\pi(y_j^-, y^*; x) - \Delta_\pi(y_j^+, y^*; x)\\ \Leftrightarrow\quad &\Delta_\pi(y_i^-, y_i^+; x) > \Delta_\pi(y_j^-, y_j^+; x),\end{aligned}$$

due to property (i) of Proposition 3.3. □

A.3 Proof of Proposition 3.4

As in (Amini et al., 2024; Azar et al., 2024), the closed-form solution of (13) is:

$$\pi(y^+|x) \propto \pi_{\text{ref}}(y^+|x) \exp\left(\beta^{-1}\mathbb{E}_{y^+\sim\pi_{\text{ref}}(\cdot|x)} p(y^+ \succ \pi_{\text{ref}}|x) + \lambda r(y^+|p \oplus x)\right).$$

Similarly for y^-, we have

$$\pi(y^-|x) \propto \pi_{\text{ref}}(y^-|x) \exp\left(\beta^{-1}\mathbb{E}_{y^-\sim\pi_{\text{ref}}(\cdot|x)} p(y^- \succ \pi_{\text{ref}}|x) + \lambda r(y^-|p \oplus x)\right).$$

After simplification, we have:

$$\begin{aligned}\frac{\pi(y^+|x)}{\pi(y^-|x)} &= \frac{\pi_{\text{ref}}(y^+|x)}{\pi_{\text{ref}}(y^-|x)} \exp\left(\beta^{-1}[\mathbb{E}_{y^+\sim\pi_{\text{ref}}(\cdot|x)} p(y^+ \succ \pi_{\text{ref}}|x) - \mathbb{E}_{y^-\sim\pi_{\text{ref}}(\cdot|x)} p(y^- \succ \pi_{\text{ref}}|x)]\right.\\ &\quad \left. +\lambda[r(y^+|p \oplus x) - r(y^-|p \oplus x)]\right).\end{aligned}$$

Note that $r(y^+|p \oplus x) - r(y^-|p \oplus x) = \Delta_\pi(y^-, y^+; x)$. After some simplifications, we have

$$\begin{aligned}&\log \frac{\pi(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \log \frac{\pi(y^-|x)}{\pi_{\text{ref}}(y^-|x)}\\ &= \frac{1}{\beta}\left[\mathbb{E}_{y^+\sim\pi_{\text{ref}}(\cdot|x)} p(y^+ \succ \pi_{\text{ref}}|x) - \mathbb{E}_{y^-\sim\pi_{\text{ref}}(\cdot|x)} p(y^- \succ \pi_{\text{ref}}|x)\right] + \lambda\Delta_\pi(y^-, y^+; x).\end{aligned}$$

As $\mathbb{E}_{y^+\sim\pi_{\text{ref}}(\cdot|x)} p(y^+ \succ \pi_{\text{ref}}|x) - \mathbb{E}_{y^-\sim\pi_{\text{ref}}(\cdot|x)} p(y^- \succ \pi_{\text{ref}}|x) = \frac{1}{2}$ (Azar et al., 2024), after taking expectations of both sides, we obtain

$$\log \frac{\pi(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \log \frac{\pi(y^-|x)}{\pi_{\text{ref}}(y^-|x)} = \frac{1}{2\beta} + \lambda\Delta_\pi(y^-, y^+; x). \tag{19}$$

B Example Cases

Tables 5, 6, 7, and 8 show four examples of responses by Sr-DPO, Sr-IPO, DPO, and IPO. As can be seen, the proposed Sr-DPO and Sr-IPO outperform DPO and IPO under GPT4's judgement.

Table 5: Example 1 for Sr-DPO and DPO. The first row is the query, the second and third rows are the responses generated by Sr-DPO and DPO, respectively, and the last row is the judgement made by GPT-4.

Prompt	Explain the process of natural selection and how it contributes to the evolution and adaptation of species.
Sr-DPO	Natural selection is the process of survival of the fittest, which occurs when organisms with the most advantageous traits are more likely to reproduce and pass on their traits than organisms with less desirable traits. It is a key factor in the evolution of species, with the most successful species being those that are the most adaptable to the environment. Natural selection ensures that organisms with the most advantageous traits are more likely to survive and pass on their traits to the next generation. Here is a brief explanation of the process of natural selection: 1. The most adaptable organisms with the most advantageous traits are more likely to survive and pass on their traits to the next generation. 2. The most advantageous traits of organisms are those that are most effective at surviving and reproducing in their environment. 3. Those organisms with the most advantageous traits are more likely to survive and reproduce than organisms with less advantageous traits.
DPO	Natural selection is the process of survival of the fittest, which occurs when the genetic makeup of a population changes over time, favoring individuals that have beneficial traits such as increased physical strength, increased intelligence, or other adaptive features. This process allows for the survival of the species that is best able to adapt to its environment.
Judgment	Both assistants provide a correct explanation of natural selection, describing it as a process that favors individuals with beneficial traits, leading to changes in the genetic makeup of a population over time. However, Assistant A's response is more detailed and comprehensive. It explains how the probability of an individual's survival and reproduction is influenced by how well adapted the organism is to its environment, and it also mentions the importance of natural selection in the field of biology and evolution. Therefore, Assistant A's response is more informative and provides a deeper understanding of the topic. Final Verdict: [[A]]

Table 6: Example 2 for Sr-DPO and DPO. The first row is the query, the second and third rows are the responses generated by Sr-DPO and DPO, respectively, and the last row is the judgement made by GPT-4.

Prompt	Draft an apology email to a customer who experienced a delay in their order, and provide reassurance that the issue has been resolved.
Sr-DPO	Here's an apology email draft for a customer who experienced a delay in their order: Dear Customer, Thank you for your patience during our recent order process. We recently experienced a delay with your order and apologize for the inconvenience. We are currently working to process your order, and we will provide a notification when it is ready. We sincerely apologize for the delay and appreciate your patience. Our team is working hard to resolve this issue as soon as possible. We look forward to welcoming your order and providing you with the best customer service experience possible. Best regards, Customer Support Team
DPO	Here's an apology email draft for a customer who experienced a delay in their order: Dear [Customer Name], apologize for the delay in your delivery of your order. I have worked with our team to resolve the issue, and I am very pleased to let you know that the issue has been resolved and your order is now ready for dispatch. I look forward to providing you with a great customer experience in the future. Sincerely, [Customer Name] Customer Support
Judgment	Both assistants provided a good response to the user's request. However, Assistant A's response was more detailed and provided more reassurance to the customer. Assistant A mentioned that they reviewed the issue with their fulfillment partners and are now shipping the order as quickly as possible. This gives the customer a sense of urgency and care for their order. On the other hand, Assistant B's response was also good but lacked the level of detail and reassurance that Assistant A's response provided. Therefore, Assistant A's response is better. Final Verdict: [[A]]

Table 7: Example 3 for Sr-IPO and IPO. The first row is the query, the second and third rows are the responses generated by Sr-IPO and IPO, respectively, and the last row is the judgement made by GPT-4.

Prompt	Why might someone choose to use a paper map or ask for directions instead of relying on a GPS device or smartphone app?
Sr-IPO	A paper map or paper printed directions can be a fun way to help you get from point A to point B. Paper maps are portable and easy to use, and they're also a fun and easy way to explore your surroundings and get around town. They're also less expensive than GPS systems or smartphone apps, and they're very easy to use, so they're a good option for older kids or adults who need to get by without a smartphone. Paper maps offer a more personal and hands-on experience than smartphone apps or GPS devices, and they can be a fun way to get around town or explore your surroundings.
IPO	A paper map or paper printed directions can be a fun way to help you get from point A to point B. Paper maps are portable and easy to use, and they're also a fun and easy way to explore your surroundings and get around town. They're also less expensive than GPS systems or smartphone apps, and they're very easy to carry around. If you are interested in using a paper map or directions, here are a few reasons why you might choose to do so:1. Paper maps are portable and easy to use. 2. Paper maps are less expensive than GPS systems or smartphone apps. 3. Paper maps are easy to carry around. 4. Paper maps are easy to use and interpret. 5. Paper maps can help you explore your surroundings and get around town. 6. Paper maps are easily outdated or outdated, so it is important to be sure you have the latest version of the map to use. 7. Paper maps are a fun way to explore your surroundings and get around town. 8. Paper maps are easy to use and interpret. 9. Paper maps can help you explore your surroundings and get around town. 10. Paper maps make it easy to get around town and explore your surroundings. 11. Paper maps are a fun and easy way to explore your surroundings and get around town. Remember to tailor the activities to the birthday childs interests and preferences. Have a great celebration!
Judgment	Assistant A's response is more concise and to the point, providing a clear explanation of why someone might choose to use a paper map or ask for directions instead of relying on a GPS device or smartphone app. Assistant B's response, while it does provide some valid points, is repetitive and confusing, especially the last sentence about tailoring activities to a birthday child's interests and preferences, which is completely irrelevant to the question asked. Therefore, Assistant A's response is more helpful and relevant. Final Verdict: [[A]]

Table 8: Example 4 for Sr-IPO and IPO. The first row is the query, the second and third rows are the responses generated by Sr-IPO and IPO, respectively, and the last row is the judgement made by GPT-4.

Prompt	How many lightning strikes occur on Earth each day? Try to explain your answer. Your explanation should take the reader through your reasoning step-by-step.
Sr-IPO	There are about 2,500 lightning strikes each day on Earth. That's about two lightning strikes for every minute! The most common type of lightning strikes occur in the tropics, where temperatures range from 70 to 90 degrees Fahrenheit. Lightning strikes are most common around noon and also around sunset. The next most common type of lightning occurs in the polar regions, where the temperatures can reach as low as -20 degrees Celsius. Lightning strikes are most common during the winter months in these regions. The most dangerous type of lightning occurs when the atmospheric pressure is extremely low, like when a storm is approaching. Low atmospheric pressure can cause the air to move faster and more violently, creating a stronger electric current through the air. When the air is moving quickly, it is also possible for an electrical charge to build up between the tips of a cloud or a thunderstorm. This electrical charge can then be released as lightning. Low atmospheric pressure can also increase the intensity of a thunderstorm, leading to more severe lightning storms.
IPO	There are about 2,500 lightning strikes each day on Earth. That's about two lightning strikes for every minute! The most common type of lightning strikes occur in the tropics, where temperatures range from 70 to 90 degrees Fahrenheit. Lightning strikes are most common around noon and also around sunset. The next most common type of lightning occurs in the polar regions, where the temperatures can reach as low as -20 degrees Celsius. Lightning strikes are most common during the winter months in these regions.
Judgment	Both Assistant A and Assistant B start their responses with the same information, stating that there are about 2,500 lightning strikes each day on Earth. They both also provide similar information about where and when lightning strikes are most common. However, Assistant A goes a step further by explaining the conditions under which the most dangerous type of lightning occurs, providing a more detailed and comprehensive response to the user's question. Therefore, Assistant A's response is more informative and thorough. Final Verdict: [[A]]

C Details on Obtaining $\pi(y|p \oplus x)$ and $r(y^+|p \oplus x) - r(y^-|p \oplus x)$

Denote the first $j-1$ tokens from y as $y_{<j} = [y_1, \ldots, y_{j-1}]$. Define $y_{<1} = \text{null}$.

The constructions of $\pi(y|p \oplus x)$ and $r(y|p \oplus x)$ are as follows. For simplicity of exposition, we assume that the mini-batch size is 1.

i. Concatenate prompt p and input x together ($p \oplus x$ = `torch.cat`([p, x])).

ii. Feed $p \oplus x$ to the model. For every token (i.e., from $j = 1$ to $j = |y|$, where $|y|$ is the length of y), obtain $\pi(y_j|p \oplus x, y_{<j})$, the probability of y_j given $p \oplus x$ and $y_{<j}$.

iii. Obtain $\pi(y|p \oplus x) = \prod_{j=1}^{|y|} \pi(y_j|p \oplus x, y < j)$.

iv. Use the same procedure to obtain $\pi_{\text{ref}}(y|p \oplus x)$, and finally obtain $r(y^+|p \oplus x) - r(y^-|p \oplus x) = \beta \log \frac{\pi_\theta(y^+|p \oplus x)}{\pi_{\text{ref}}(y^+|p \oplus x)} - \beta \log \frac{\pi_\theta(y^-|p \oplus x)}{\pi_{\text{ref}}(y^-|p \oplus x)}$.