# Indirect Online Preference Optimization via Reinforcement Learning

**En Wang[1,2], Xingyu Lin[1,2], Du Su[3], Chenfu, Bao[4],**
**Zhonghou Lv[4], Funing Yang[1,2], Yuanbo Xu[1,2], Wenbin Liu[1,2*]**
[1]College of Computer Science and Technology, Jilin University
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering of MOE, Jilin University
[3]State Key Laboratory of AI Safety, Institute of Computing Technology, Chinese Academy of Sciences
[4] Baidu Inc.
{wangen,linxy23,yfn,yuanbox,liuwenbin}@jlu.edu.cn,
sudu@ict.ac.cn, {baochenfu, lvzhonghou}@baidu.com,

## Abstract

Human preference alignment (HPA) aims to ensure Large Language Models (LLMs) responding appropriately to meet human moral and ethical requirements. Existing methods, such as RLHF and DPO, rely heavily on high-quality human annotation, which restrict the efficiency of iterative online model refinement. To address the inefficiencies of human annotation acquisition, iterated online strategy advocates the use of fine-tuned LLMs to self-generate preference data. However, this approach is prone to distribution bias, because of differences between human and model annotations, as well as modeling errors between simulators and real-world contexts. To mitigate the impact of distribution bias, we adopt the principles of adversarial training, framing a zero-sum two-player game with a protagonist agent and an adversarial agent. With the adversarial agent challenging the alignment of protagonist agent, we continuously refine the protagonist's performance. By utilizing minmax equilibrium and Nash equilibrium strategies, we propose Indirect Online Preference Optimization (IOPO) mechanism that enables the protagonist agent to converge without bias while maintaining linear computational complexity. Extensive experiments across three real-world datasets demonstrate that IOPO outperforms state-of-the-art alignment methods in both offline and online scenarios, evidenced by standard alignment metrics and human evaluations. This innovation reduces the time required for model iterations from months to one week, alleviates distribution shifts, and significantly cuts annotation costs.

## 1 Introduction

LLMs, such as GPT-4 [Achiam *et al.*, 2023] and Baichuan-2 [Yang *et al.*, 2023], exhibit impressive reasoning capabilities and advanced functionalities. However, a significant challenge arises in how to ensure that LLMs respond appropriately in special domains to meet human moral and ethical requirements, particularly in politics and racial affairs related to safety. Adjusting the bias of LLMs to align with human preferences is referred to as Human preference alignment[Ouyang *et al.*, 2022].

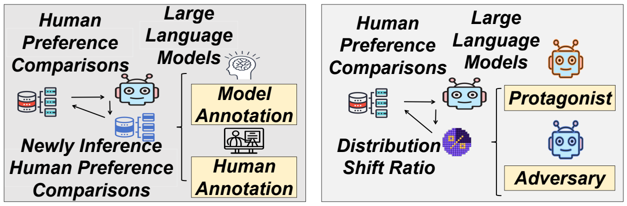


Figure 1: ***Fine-Tuning Pipeline (left) vs. IOPO Pipeline (right).*** IOPO optimization without requiring continuous annotation.

To address the above challenges, researchers have explored various algorithms to inject desired behaviors into LLMs. Reinforcement learning from human feedback (RLHF) [Bai *et al.*, 2022a] advocates for training multiple LLMs while using a reward model to score policy learning. Rafailov $et\ al.$ propose a Direct Preference Optimization (DPO) algorithm reducing reliance on reward models[Rafailov *et al.*, 2024]. However, all methods encounter a significant bottleneck: the need for high-quality human preference comparisons, which consists of paired model outputs with human annotations indicating preference alignment.

For the challenge of this inefficient acquisition of high-quality preference comparisons, iterated online strategy advocates the use of fine-tuned LLMs, to construct preference data through self-questioning and self-answering periodically [Ouyang *et al.*, 2022; Bai *et al.*, 2022b; Bai *et al.*, 2022a; Rafailov *et al.*, 2024; Xu *et al.*, 2024]. However, this framework suffers from distribution bias. On one hand, there are differences between human annotation and model annotation. Touvron $et\ al.$[Touvron *et al.*, 2023] note that sampling distribution shift in online procedure incurs these differences and modeling errors: a distribution gap emerges between model-generated preference data and human annotation preference data after a few iterations. On the other hand, any established

[*] Corresponding Author

LLMs (simulator) may differ from the real environment. If the learned policies are not robust enough to account for modeling errors, transferring these policies from the simulator to the real world often fails [Pinto *et al.*, 2017]. Therefore, although LLMs that have gone fine-tuned LLMs own extensive vertical domain knowledge, high-quality preference data annotated by them often fail to generalize to real scenarios.

We propose **Indirect Online Preference Optimization (IOPO)** to address distributional bias without additional annotations. **Theoretical Foundations**: Our dual-agent architecture employs *inverse reward design* within a rigorously formalized *two-player zero-sum game* framework [Littman, 1994]. Through joint optimization of both agents' reward functions, the system guarantees convergence to Nash equilibrium with provable stability. **Symmetrically-Constrained Bias Mitigation**: The Nash equilibrium in our framework naturally counteracts bias via adversarial dynamics. Building upon RARL [Pinto *et al.*, 2017], our zero-sum constraints effectively recast the agent's bias as adversarial perturbations, thereby achieving systematic bias reduction. The key innovation lies in our symmetric reward formulation, which unlike conventional adversarial training, imposes balanced constraints on both agents' reward structures. This prevents pathological collapse into biased strategies while maintaining the adversarial agent's role in continuously refining the protagonist's policy alignment. The resulting IOPO mechanism ensures bias-free convergence while preserving computational efficiency with linear complexity.

Our framework necessitates careful consideration of two key challenges: (1) dual-agent computational overhead and (2) convergence instability in zero-sum games. We reformulate the Bradley-Terry model via importance sampling (IS) [Tokdar and Kass, 2010] to quantify adversarial influence, and propose a clipped IS scheme that ensures: (i) proximal policy approximation, (ii) bounded policy trajectories, while preserving IS benefits (efficiency, policy compatibility, unbiasedness). This achieves reliable adversarial optimization.

In summary, this paper makes the following contributions:

- We propose a novel indirect online preference optimization (ORL) framework that simulates the real environment with no bias, without increasing annotation costs.
- In light of excessive computational resource consumption and slow-or-no convergence problem in ORL, we impose a simple clipping on IS weight to perform proximal approximation calculations.
- Our approach outperforms the SOTA approach method DPO in three real world datasets. Online evaluations validate that the iterated online strategy with few human annotations exhibits issues related to distribution shift in online preference optimization, while our method reduces thousands of expenses and time in month in annotation process.

## 2 Related Work

### 2.1 Reinforcement learning on Markov

The Markov process of standard reinforcement learning (RL) is represented by a tuple $(AGENT, ENV) \equiv (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$. Here, $AGENT$ is the entity making decisions and learning, $ENV$ is the external system. The actions of $AGENT$ influence the environment, thereby allowing the $AGENT$ to receive feedback. $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is the transition probability, $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward, $\gamma$ is the discount factor. Main procedure of $AGENT$, such as Actor-Critic [Konda and Tsitsiklis, 1999] and PPO [Schulman *et al.*, 2017], is to learn a stochastic policy $\pi_\theta : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ that maximizes the cumulative discounted reward $\sum_{t=0}^{T-1} \gamma^t r(s_t, a_t)$, where $\theta$ denotes the parameters for the policy $\pi$ at time-step $t$. Levine *et al.* [Levine *et al.*, 2020] delineate key distinctions: online RL involves $ENV$ - $AGENT$ iterations absent in offline RL, while on-policy RL unifies behavior and learning in a single policy versus off-policy's separation.

Extending RARL [Christiano *et al.*, 2017], we introduce a clipped IS-weighted zero-sum game to reduce HPA bias.

### 2.2 Human Preference Fine-Tuning

Human preference alignment [Ouyang *et al.*, 2022] seeks to align LLMs with human expectations. Notably, Bai $et\ al.$ introduce Reinforcement Learning from Human Feedback (RLHF) to develop a helpful and harmless assistant [Bai *et al.*, 2022a], utilizing self-questioning and self-answering techniques [Bai *et al.*, 2022b]. RLHF faces challenges, including significant computational costs, complex complementary procedures, and instability due to reward hacking. To enhance sampling efficiency and stability, Rafailov $et\ al.$ propose Direct Preference Optimization (DPO) as a means to bridge the gap between reward functions and policy [Rafailov *et al.*, 2024]. RSO [Liu *et al.*, 2023] improves the estimation of the optimal policy by integrating SLiC with DPO. Addressing overfitting and generalization issues, IPO [Azar *et al.*, 2024] mathematically critiques DPO's limitations and suggests a comprehensive objective for learning from human preferences. KTO-PAIR [Ethayarajh *et al.*, 2024] advances this by directly optimizing human-aware losses (HALOs) instead of merely maximizing the log-likelihood of preferences, using a Kahneman-Tversky model of human utility. Nonetheless, most existing fine-tuning methods pose a need for high-quality human preference comparisons, which restricts continuous model updates, underscoring the need for further research in this domain.

Recently, iterated online strategy has emerged as a solution for continuous fine-tuning [Ouyang *et al.*, 2022; Bai *et al.*, 2022b; Bai *et al.*, 2022a; Rafailov *et al.*, 2024; Xu *et al.*, 2024]. Self-Reward [Yuan *et al.*, 2024] proposes that LLMs utilize a Judge-Prompt mechanism to generate their own rewards during training. Iter-FineTuning [Xiong *et al.*, 2024] introduces an iterative version of Direct Preference Optimization for online settings and a multi-step rejection sampling strategy for offline scenarios, effectively enhancing policy learning. APO [Cheng *et al.*, 2024] suggests an alternating update approach for LLMs and reward models through a min-max game using high-quality data. While existing online methods have partially mitigated the limitations of offline approaches, sampling distribution issues continue to impede alignment efficiency [Touvron *et al.*, 2023], necessitating additional human annotations. For instance, DPO's effective-

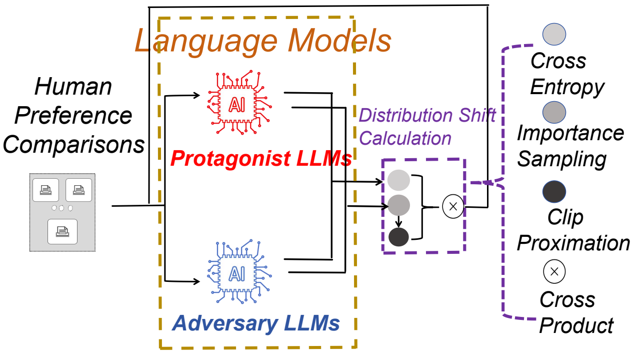


Figure 2: **The overall structure of IOPO.** The model architecture integrates human preference comparisons as inputs. During $N_\mu$ iterations, adversarial LLMs generate a softmax output $\pi_\nu(y^t \mid x^t)$, which enables the protagonist LLMs to compute a binary cross-entropy loss and a clipped importance sampling (IS) weight with protagonist's softmax output. The product of these two components is expressed as an objective function Eq.(14). In the following $N_\nu$ iterations, the adversarial LLMs analogously compute a corresponding quantity derived from the protagonist LLMs' softmax output.

ness is contingent upon high-quality data, and RLH(A)F reward models require real-time updates.

Our dual-agent architecture implements *inverse reward design* within a formal *two-player zero-sum game* framework [Littman, 1994], guaranteeing Nash equilibrium convergence with provable stability. Extending with [Pinto *et al.*, 2017], we develop a bias-reducing zero-sum two-player game.

## 3 System Model and Problem Formulation

The fine-tuning pipeline consists of three stages: 1) unsupervised pretraining of LLMs, 2) LLMs aligned through supervised fine-tuning (SFT), and 3) LLMs optimized with RL algorithms, commonly REINFORCE [Zoph, 2016], proximal policy optimization (PPO) [Schulman *et al.*, 2017], DPO [Rafailov *et al.*, 2024], or their variants.

For DPO, given a tuple $(x, y_l, y_w)$ sampled from dataset $\mathcal{D} = \left\{x^{(i)}, y_w^{(i)}, y_l^{(i)}\right\}_{i=1}^{N}$, the preference is denoted as $y_w \succ y_l \mid x$, where $y_w$ and $y_l$ represent the preferred and less preferred completions, respectively.

$$Q\left(y_w \succ y_l \mid x\right) = \frac{\exp\left(r_\phi\left(x, y_w\right)\right)}{\exp\left(r_\phi\left(x, y_w\right)\right) + \exp\left(r_\phi\left(x, y_l\right)\right)} \quad (1)$$
$$= \log \sigma(r_\phi(x, y_l) - r_\phi(x, y_w)).$$

where $Q\left(y_w \succ y_l \mid x\right)$ or $P\left(y_w \succ y_l \mid x\right)$ calculated based on the Bradley-Terry mathematical model[Bradley and Terry, 1952]. The optimization of the reward model $r_\phi(x, y)$ can be formulated as maximizing the negative log-likelihood loss:

$$\mathcal{L}_R(r_\phi, \mathcal{D}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma(r_\phi(x, y_l) - r_\phi(x, y_w)\right]. \quad (2)$$

Following policy optimization problem as formulating:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}\left[r_\phi(x, y)\right] - \beta \mathbb{D}_{\mathbf{KL}}\left[\pi_\theta(y \mid x) \mid\mid \pi_{\text{ref}}(y \mid x)\right], \quad (3)$$

where $\beta$ is a hyper parameter, controlling the deviation from model policy $\pi_\theta$ to reference policy $\pi_{\text{ref}}$, with $\pi_\theta$ is also initialized to $\pi^{\text{SFT}}$.

To simplify the pipeline, motivated by [Rafailov *et al.*, 2024], we deduce the policy $\pi_\theta$ from the objective function:

$$\pi_r(y \mid x) = \frac{1}{Z(x)} \pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right),$$
$$Z(x) = \sum_y \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right), \quad (4)$$

where we rename the derivation outcome as $\pi_r(y \mid x)$. Subsequently, the reward model $r(x, y)$ is presented as follows:

$$r(x, y) = \beta \log \frac{\pi_r(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x). \quad (5)$$

Substituting Eq.(5) into Eq.(1), here is reward function:

$$Q\left(y_w \succ y_l \mid x\right) = \frac{1}{1+\exp\left(\beta \log \frac{\pi(y_l|x)}{\pi_{\text{ref}}(y_l|x)} - \beta \log \frac{\pi(y_w|x)}{\pi_{\text{ref}}(y_w|x)}\right)}. \quad (6)$$

Therefore, policy objective and the gradient of the loss function are shown below:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}$$
$$\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right], \quad (7)$$

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\beta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}$$
$$\left[\sigma(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w))\left[\nabla_\theta \log \pi(y_w \mid x) - \nabla_\theta \log \pi(y_l \mid x)\right]\right], \quad (8)$$

where $\hat{r}_\theta(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$, denoting reward implicitly defined by $\pi_\theta(y|x)$ and $\pi_{\text{ref}}(y|x)$.

## 4 Methodology

The original fine-tuning as Fig.1 (left) poses additional annotation cost caused by sampling distribution shifts: LLMs, via DPO (RLHF), will infer new Preference Comparison Completions to continuously update their models or reward models. In contrast, we employ a two-players zero-sum game to calculate distribution shift ratio as Fig.1 (right).

### 4.1 Formulations of Indirect Online Preference Optimization

Both players observe the state $s_t$ and take actions $a_t^\mu \sim \mu(s_t)$ or $a_t^\nu \sim \mu(s_t)$, where $\mu$ and $\nu$ represent the strategies of protagonist and adversary, respectively. We note that $a_t^\mu \sim \mu(s_t)$ and $a_t^\nu \sim \mu(s_t)$ denote preferences as $y_w \succ y_l \mid x$ and $y_l \succ y_w \mid x$. The state transitions $s_{t+1} = P(s_t, a_t^\mu, a_t^\nu)$ are given by $\pi(\mu, \nu)$, and a reward $r_t = r(s_t, a_t^\mu, a_t^\nu)$ is defined as $r_t^\mu = r_t$ while the adversary gets a reward $r_t^\nu = -r_t$. Reward function optimization is evaluated as follows:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\mu(y|x)}\left[r_\phi(x, y)\right] - \beta \mathbb{D}_{\mathbf{KL}}\left[\pi_\mu(y \mid x) \mid\mid \pi_\nu(y \mid x)\right]. \quad (9)$$

As for computation costs, $r^t(\theta)$ quantifies the distribution shift caused by protagonist $\pi_\mu$ and adversary $\pi_\nu$. RARL [Pinto *et al.*, 2017] demonstrated that the notions of minmax equilibrium and Nash equilibrium are equivalent for this game, with an optimal equilibrium reward denoted as $Q^{\mu*}$.

$$Q^{u*} = \min_\nu \max_\mu Q^u(\mu, \nu) = \max_\mu \min_\nu Q^u(\mu, \nu). \quad (10)$$

Similarly, the adversary aims to maximize its own reward through $Q^{\nu} \equiv Q^{\nu}(\mu, \nu) = -Q^{\mu}(\mu, \nu)$. And we observe that $\pi_{\mu}(y_l^t \mid x^t) \equiv \pi_{\nu}(y_w^t \mid x^t)$:

$$r_{\mu}^t(\theta) = \begin{cases} 1 & \text{, if } r_{\mu}^t(\theta) = 0 \text{ or } \infty, \\ \frac{\pi_{\mu}(y_l^t|x^t)}{\pi_{\nu}(y_l^t|x^t)} = \frac{\pi_{\nu}(y_w^t|x^t)}{\pi_{\nu}(y_l^t|x^t)} & \text{, other case ,} \end{cases} \tag{11}$$

We should emphasize that $r_{\mu}^t(\theta) = 0$ represents overfitting in pipeline, while $r_{\mu}^t \to \infty$ situations where learning the data features is difficult.

Due to the slow-or-no convergence problem, we employ a simple clip proximal calculation on $r_{\mu}^t(\theta)$ to penalize significant deviations staying away from the ratio of norm.

$$r_{\mu}^t(\theta) = \min(r_{\mu}^t(\theta), \text{clip}(r_{\mu}^t(\theta), 1-\epsilon, 1+\epsilon)) = -r_{\nu}^t(\theta). \tag{12}$$

Given the constraint $\sum_y \pi(y|x) = 1$, we perform Lagrange differentiation on Eq.(9) to derive the policy $\pi_{\mu}$:

$$\begin{aligned} &\mathcal{Q}_{\mu}^t\left(y_w \succ y_l \mid x\right) = -\mathcal{Q}_{\nu}^t\left(y_w \succ y_l \mid x\right) \\ =&\frac{1}{r_{\mu}^t(\theta) * (1 + \exp\left(\beta \log \frac{\pi_{\mu}^t(y_l|x)}{\pi_{\nu}^t(y_l|x)} - \beta \log \frac{\pi_{\mu}^t(y_w|x)}{\pi_{\nu}^t(y_w|x)}\right))}. \end{aligned} \tag{13}$$

Therefore, using Eq.(13), we can transform Eq.(7) into Eq.(14) and Eq.(8) into Eq.(15), respectively.

$$\begin{aligned} &\mathcal{L}_{\text{IOPO}}(\pi_{\mu}; \pi_{\nu}) = -\mathbb{E}_{(x, y_{\omega}, y_1)\sim\mathcal{D}}^t \\ &\left[r^t(\theta) * \log\sigma\left(\beta\log\frac{\pi_{\mu}^t(y_w|x)}{\pi_{\nu}^t(y_w|x)} - \beta\log\frac{\pi_{\mu}^t(y_w|x)}{\pi_{\nu}^t(y_w|x)}\right)\right], \end{aligned} \tag{14}$$

$$\begin{aligned} &\nabla_{\mu}\mathcal{L}_{\text{IOPO}}(\pi_{\mu}; \pi_{\nu}) = -\beta\mathbb{E}_{(x, y_w, y_l)\sim\mathcal{D}} \\ &\left[r_{\mu}^t * \log\sigma\left(\beta\log\frac{\pi_{\mu}^t(y_w|x)}{\pi_{\nu}^t(y_w|x)} - \beta\log\frac{\pi_{\mu}^t(y_l|x)}{\pi_{\nu}^t(y_l|x)}\right)\right] \\ &[\nabla_{\pi_{\mu}}\log\pi_{\mu}(y_w \mid x) - \nabla_{\pi_{\mu}}\log\pi_{\mu}(y_l \mid x)], \end{aligned} \tag{15}$$

which consist of the objective function and its backward derivative formula for the protagonist.

### 4.2 Indirect Online Preference Optimization Outline

The model structure is illustrated in Fig.2, where human preference comparisons inform our models. Over $N_{\mu}$ iterations, adversarial LLMs supply the protagonist LLMs with a softmax output $\pi_{\nu}(y^t \mid x^t)$. This output is used by the protagonist LLMs to compute a binary cross-entropy loss and an IS weight with clipping. The product of the cross-entropy loss and the clipped IS weight forms the foundation of the reverse iteration process, as detailed in Eq.(15). During $N_{\nu}$ iterations, the adversarial LLMs derive the cross-product of the binary cross-entropy loss and the clipped IS weight from the softmax output of the protagonist LLMs.

Pipeline is as Algorithm.1: 1) initialize policy models $\mu$ and $\nu$ via SFT; 2) In practice, we carry out an iterative two-step optimization procedure. First, for $N_{\mu}$ iterations, the parameters of the adversary $\theta_{\nu}^{t-1}$ are held constant and serve as the $ENV$, while the parameters $\theta_{\mu}^t$ of the protagonist $AGENT$ are optimized to maximize the reward function Eq.(14). For the next step of $N_{\nu}$ iterations, the parameters of the protagonist are held constant, and the parameters of the adversary $\theta_{\nu}^t$ are optimized via Eq.(15).

**Algorithm 1** Indirect Online Preference Optimization

```
Input: Datasets $\mathcal{D} = \left\{x^{(i)}, y_w^{(i)}, y_l^{(i)}\right\}_{i=1}^N$
Parameter: $\theta_{\mu}^0$ for $\mu$ and $\theta_{\nu}^0$ for $\nu$
Output: $\theta_{\mu}^{N_{iter}}$
for $i \leftarrow 1$ to $N_{iter}$ do
    $\theta_{\mu}^i \leftarrow \theta_{\mu}^{i-1}$
    for $j \leftarrow 1$ to $N_{\mu}$ do
        $\theta_{\mu}^i$ and $\theta_{\nu}^{i-1}$ serve as $AGENT$ and $ENV$
        $\theta_{\mu}^i \leftarrow$ Optimization with (15)
    End For

    $\theta_{\nu}^i \leftarrow \theta_{\nu}^{i-1}$
    for $j \leftarrow 1$ to $N_{\nu}$ do
        $\theta_{\nu}^i$ and $\theta_{\mu}^i$ serve as $AGENT$ and $ENV$
        $\theta_{\mu}^i \leftarrow$ Optimization with (15)
    End For

End For
return $\theta_{\mu}^{N_{iter}}$
```

## 5 Theoretical Analysis

**Theorem 1.** *When $t \to \infty$, our $Q(\mu, \nu)$ that is a zero-sum two-player game, is monotonic and converge.*

*Proof.* When $t \to \infty$, $Q(\mu, \nu)$ is a zero-sum game as follow:

$$\begin{aligned} (Q_{\mu}^t - Q_{\nu}^{t-1}) + (Q_{\nu}^t - Q_{\mu}^t) = 0, \\ (Q_{\mu}^t - Q_{\nu}^{t-1}) + (Q_{\nu}^{t-1} - Q_{\mu}^{t-1}) = 0, \end{aligned} \tag{16}$$

where $Q_{\mu}^t$ denotes mean-value of protagonist at $t$ iteration. Therefore, we deduce from Eq.(16) to Eq.(17):

$$(Q_{\nu}^t - Q_{\nu}^{t-1}) = (Q_{\mu}^t - Q_{\mu}^{t-1}). \tag{17}$$

Similarly, there is $(Q_{\nu}^t - Q_{\nu}^{t-1}) = (Q_{\mu}^{t+1} - Q_{\mu}^t)$. By this equation and Eq.17, we have :

$$(Q_{\mu}^t - Q_{\mu}^{t-1}) = (Q_{\mu}^{t+1} - Q_{\mu}^t). \tag{18}$$

Eq.(18) indicates that the reward function $Q$ forms an arithmetic progression. Given that $Q \in range(Q_{\mu}^t * (1/r^t(\theta)))$, where $Q_{\mu}^t \in [0, 1]$ and $r^t(\theta)$ is constant, we conclude that our algorithm converges based on Theorem (1). □

**Theorem 2.** *With $r^t(\theta)$ is constant, our $Q_{\mu}$ converge faster.*

*Proof.* We note that for $N_{\mu}$ iteration:

$$\begin{aligned} &Q_{\mu}^t - Q_{\mu}^{t-1} = \mathcal{Q}_{\mu}^t\left(y_w \succ y_l \mid x\right) \\ &\propto (1/r_{\mu}^t(\theta)) * \left(\beta\log\frac{\pi_{\mu}^t(y_l|x)}{\pi_{\nu}^t(y_l|x)} - \beta\log\frac{\pi_{\mu}^t(y_w|x)}{\pi_{\nu}^t(y_w|x)}\right) \\ &\propto (1/r_{\mu}^t(\theta)) * (\pi_{\mu}^t(y_w|x)/\pi_{\mu}^t(y_l|x)), \end{aligned} \tag{19}$$

where $r_{\mu}^t(\theta)$ is constant in $N_{\mu}$ iteration.

Transforming Eq.(18) with Eq.(19) into Eq.(20):

$$\begin{aligned} &(Q_{\mu}^t - Q_{\mu}^{t-1})/(Q_{\mu}^{t+1} - Q_{\mu}^t) \\ &= (\pi_{\nu}^{t+1}(y_w|x)/\pi_{\nu}^{t+1}(y_l|x))/(\pi_{\nu}^t(y_w|x)/\pi_{\nu}^t(y_l|x)) \\ &* (\pi_{\mu}^t(y_w|x)/\pi_{\mu}^t(y_l|x))/(\pi_{\mu}^{t+1}(y_w|x)/\pi_{\mu}^{t+1}(y_l|x)). \end{aligned} \tag{20}$$

| Datasets | Baselines | Win rate (IOPO vs.) by Ziya | Win rate (IOPO vs.) by ERNIE-3.5 |
|---|---|---|---|
| *safe world view* | *SFT* | - | 44.8% vs. 21.6% vs. 33.6% |
| | *DPO* | - | 50.4% vs. 21.2% vs. 28.4% |
| | *HING* | - | 49.2% vs. 20% vs. 30.8% |
| | *IPO* | - | 44.8% vs. 22% vs. 33.2% |
| | *KTO-PAIR* | - | 48% vs. 21.6% vs. 30.4% |
| *distilabel-capybara-dpo* | *SFT* | 13%vs. 75%vs.12% | 65% vs. 3% vs. 32% |
| | *DPO* | 12%vs. 78% vs. 10% | 50% vs. 8% vs. 42% |
| | *RSO* or HINGE | 10% vs. 78% vs. 12% | 53%vs. 18% vs. 39% |
| | *IPO* | 12%vs. 76% vs. 12% | 51% vs. 7% vs. 42% |
| | *KTO-PAIR* | 12% vs. 78% vs. 10% | 48% vs. 7% vs. 46% |
| *orpo-dpo-mix-4ok* | *SFT* | 65%vs.0%vs.35% | 48% vs. 42% vs. 10% |
| | *DPO* | 53% vs. 0% vs. 47% | 50% vs. 34% vs. 16% |
| | *HING* | 48% vs. o% vs. 52% | 48% vs. 36% vs. 16% |
| | *IPO* | 54% vs. 0% vs. 46% | 46% vs. 32% vs. 22% |
| | *KTO-PAIR* | 65% vs. o% vs. 35% | 45% vs. 35% vs. 20% |

Table 1: **Our method outperforms the baselines with *Baichuan2-7B-chat* in offline scenarios.** The win rate is calculated by comparing IOPO against the baselines (by *Ziya and ERNIE-3.5*) using the metrics ($R_{\text{Win}}$ vs $R_{\text{Tie}}$ vs $R_{\text{Lose}}$ ). All evaluations are performed on out-of-distribution completions at a sampling temperature of 0.8.

First, the ratio $(\pi^t_\mu(y_w|x)/\pi^t_\mu(y_l|x))$ resembles a function $f(k) = k/(1-k)$, where $\pi^t_\mu(y_w|x)$ or $k \in [0,1]$. The series Newton-Leibniz formula [Bos, 1980] expansion of $f(k)$ is $k + k^\Lambda 2 + k^\Lambda 3 + ...$ and the derivation of $f(k)$, for $k \in [0,1]$ is positive. Our evaluations show that $1/2 < \pi^t_\mu(y_w|x) <= \pi^{t+1}_\mu(y_w|x)$ is generally true, implying that $(\pi^t_\mu(y_w|x)/\pi^t_\mu(y_l|x))$ increases as $\pi^t_\nu(y_w|x)$ increases. we conclude that: $\frac{\pi^t_\mu(y_w|x)\pi^{t+1}_\mu(y_l|x)}{\pi^t_\mu(y_l|x)\pi^{t+1}_\mu(y_w|x)} \leq 1$

Second, the ratio $\frac{\pi^{t+1}_\nu(y_w|x)/\pi^{t+1}_\nu(y_l|x)}{\pi^t_\nu(y_w|x)/\pi^t_\nu(y_l|x)}$ remains constant. Additionally, $\frac{\pi^{t+1}_\nu(y_w|x)/\pi^{t+1}_\nu(y_l|x)}{\pi^t_\nu(y_w|x)/\pi^t_\nu(y_l|x)} \leq 1$, similar to the aforementioned inequality, but with the opposite inequality sign when $\pi^t_\nu(y_w|x) \geq \pi^{t+1}_\nu(y_w|x)$.

In conclusion, $(Q^t_\mu - Q^{t-1}_\mu)/(Q^{t+1}_\mu - Q^t_\mu) <= 1$ denotes that $Q_\mu$ converge faster with $r^t(\theta)$. □

**Theorem 3.** *As for sampling errors[Touvron* et al.*, 2023], with high probability $\geq 1-\delta$, in each iteration of $N_{iter}$, for all $(x, y_w, y_l) \in \mathcal{D}$ , our reward function Q have a bound*

$$\begin{aligned}\hat{Q}(y_w \succ y_l \mid x) \leq Q(y_w \succ y_l \mid x) + (1/r^t(\theta)) \\ * \sigma(\frac{C_{r,delta_1}}{\sqrt{D}}) + \frac{C_{T,delta_2}}{\sqrt{D}} * (\frac{R}{1-\alpha})\end{aligned},$$

*where $C_{r,delta_1}$ is a constant dependent on the concentration properties (variance) of $r(x, y_w, y_l)$, $C_{T,delta_2}$ is a constant dependent on the concentration properties (variance) of $T(x^{t+1}|x^t, y^t_w, y^t_l)$, $\alpha = \prod_{t=1}^N (1/r^t(\theta))$ and $|Q(y_w > y_l|x)| \leq R(\forall(x, y_w, y_l))$.*

*Proof.* We know that $|r(s,a)| \leq R(\forall(s,a))$ and then $Q(s,a) \leq \frac{R}{1-\gamma}$ with Bellman Iteration[Kakade and Langford, 2002] . We deduce that $Q_\mu(y_w \succ y_l \mid x) \leq \frac{R}{1-\alpha}$, where $|Q(y_w > y_l|x)| \leq R(\forall(x, y_w, y_l))$ and $\alpha = \prod_{t=1}^N (1/r^t(\theta))$. For $i$ iteration, $r(\theta)) < 1$, we have $\prod_{t=1}^N (1/r^t(\theta)) <= \prod_{t=i}^N (1/r^t(\theta))$.

Same as SAC [Haarnoja *et al.*, 2018] or Conservative Q-Learning for Offline Reinforcement Learning[Kumar *et al.*, 2020], for all $(x, y_w, y_l) \in \mathcal{D}$ , with high probability$\geq 1-\delta_1$,

$$\begin{aligned}&|\hat{r}_\theta(x, y_w) - \hat{r}_\theta(x, y_l)| = |\hat{r}_\theta(x, y_w) - r_\theta(x, y_w)| + |\hat{r}_\theta(x, y_l) - r_\theta(x, y_l)| \\ &< 2 * \max(|\hat{r}_\theta(x, y_w) - r_\theta(x, y_w)|, |\hat{r}_\theta(x, y_l) - r_\theta(x, y_l)|) \leq \frac{C_{r,delta_1}}{\sqrt{\mathcal{D}}},\end{aligned} \tag{21}$$

where $C_{r,delta_1}$ is a constant dependent on the concentration properties of $r(x, y_w, y_l)$ and $\delta_1 \in (0,1)$, $\hat{r}_\theta(x,y) = \beta \log \frac{\pi_\mu(y|x)}{\pi_\nu(y|x)}$, $\hat{r}_\theta(x,y)$ and $r_\theta(x,y)$ denote empirical distribution function and iterative stable empirical distribution function. Motivated by Eq.(21), we have

$$|\hat{Q}^t(y_w \succ y_l \mid x - Q^t(y_w \succ y_l \mid x)| \leq (1/r^t(\theta)) * \sigma(\frac{C_{r,delta_1}}{\sqrt{\mathcal{D}}}), \tag{22}$$

where $\sigma$ is sigmoid function. Similarly as Eq.(21):

$$|\hat{T}(x^{t+1}|x^t, y^t_w, y^t_l) \quad - T(x^{t+1}|x^t, y^t_w, y^t_l)| \leq \frac{C_{T,delta_2}}{\sqrt{\mathcal{D}}}, \tag{23}$$

where $C_{T,\delta_2}$ is a constant related to the concentration properties of $T(x^{t+1}|x^t, y^t_w, y^t_l)$ with $\delta_2 \in (0,1)$. $\hat{T}(x^{t+1}|x^t, y^t_w, y^t_l)$ and $T(x^{t+1}|x^t, y^t_w, y^t_l)$ represent the empirical and iterative stable empirical distribution function of the next action under policy $\pi(\mu, \nu)$ at state $(x, y_w, y_l)$.

Here are deductions with high probability $\geq 1-\delta$:

$$\begin{aligned}&|\hat{Q}^{t+1}(y_w \succ y_l \mid x - Q^{t+1}(y_w \succ y_l \mid x)| \\ &= |\hat{Q}^t(y_w \succ y_l \mid x - Q^t(y_w \succ y_l \mid x) \\ &+ \sum_{x_{t+1}} (\hat{T} - T) E_{x^{t+1} \sim \pi}[Q^{t+1}(y_w \succ y_l \mid x)| \\ &\leq |\hat{Q}^t(y_w \succ y_l \mid x - Q^t(y_w \succ y_l \mid x)| \\ &+ |\sum_{x_{t+1}} (\hat{T} - T) E_{x^{t+1} \sim \pi}[\frac{R}{1-\alpha}]| \\ &\leq (1/r^t(\theta)) * \sigma(\frac{C_{r,delta_1}}{\sqrt{D}}) + \frac{C_{T,delta_2}}{\sqrt{D}} * (\frac{R}{1-\alpha}).\end{aligned} \tag{24}$$

□

| Baselines | Test set of ***safe world view*** | | ***250 pieces real-life online data*** | |
|---|---|---|---|---|
| | Margin reward | Pairwise accuracy | Win rate by ERNIE-3.5 | Mean value scored by human |
| *SFT* | - | - | 13 : 204 : 10 : 23 | 2.156 |
| *DPO* | 23.75 | 0.9286 | 13 : 203 : 11 : 23 | 2.164 |
| *Iter-DPO* | 7.6875 | 0.961 | 13 : 204 : 11 : 22 | 2.152 |
| *IOPO* | 11.875 | **0.9659** | - | **2.176** |

Table 2: **Our method outperforms Iter-DPO in online scenarios.** Based on human evaluation, the win rate is calculated by comparing IOPO against the baselines categorized as ($R_{\text{Win}}$ vs $R_{\text{Tie}}$ vs $R_{\text{Lose}}$ vs *bad case*). Here, a *bad case* is defined as a scenario where both answers evaluated by humans receive a score of 0. All evaluations are conducted on out-of-distribution data at a sampling temperature of 0.

# 6 Experiments

Our evaluations indicate that it performs well in assessing response quality for single-turn dialogues in single settings, multi-turn dialogues across multiple domains, and for both single-turn and multi-turn dialogues in multiple domains.

## 6.1 Datasets & Setting

There are three real-world data sets for the evaluation:

- *safe world view* contains 44k single-turn comparisons about world view demands, which is a real-world dataset sourced from Baidu.
- *distilabel-capybara-dpo* contains 7.56k multi-turn comparison data spanning math, medicine, computer science, history, and literature, among others. [1].
- The *orpo-dpo-mix-40k* dataset contains 44.2k single and multi-turn comparisons across math, medicine, CS, history, and literature. It includes *toxic-dpo-v0.2* to train models to reject illegal queries. Created via rule-based filtering (removing 2,206 GPTisms from *argilla*, *unalignment*, *M4-ai*, and *jondurbin*), it's suitable for high-quality ORPO/DPO training.[2]

We use *Baichuan2-7B* (base/chat) with standard SFT parameters. Training: learning rate $lr = 2 * \exp^{-7}$, batchsize $= 32$, $\beta = 0.1$. IOPO matches DPO's sensitivity to hyperparameters except clipping, and is robust to $\epsilon$(set to 0.3).

## 6.2 Comparison Algorithms & Metrics

To provide a comprehensive evaluation of our proposed method, we have designed a series of experiments involving several methods, as outlined below [3]:

- *DPO* [Rafailov *et al.*, 2024]: Given preference data, DPO fits a binary classifier based on the Bradley-Terry model. DPO propose a sigmoid loss on the normalized likelihood by fitting a logistic log-sigmoid function.
- *RSO* or *HINGE* [Liu *et al.*, 2023]: RSO proposes using a hinge loss on the normalized likelihood derived from the methodologies of SLiC and DPO.
- *IPO* [Azar *et al.*, 2024]: IPO is also based on the reciprocal of the gap between the log-likelihood ratios of the chosen versus rejected completion pairs.
- *KTO-PAIR* [Ethayarajh *et al.*, 2024]: Using a Kahneman-Tversky model of human utility, KTO-PAIR directly optimizes with human-aware losses (HALOs).
- *SFT*: The SFT model is optimized by training on datasets $\mathcal{D} = \left\{(x^{(i)}, y_w^{(i)})\right\}_{i=1}^{N}$.
- *Iter-DPO*: Building on prior works [Ouyang *et al.*, 2022; Yuan *et al.*, 2024; Xiong *et al.*, 2024; Cheng *et al.*, 2024], we employ LLMs (via DPO tuning) to generate preference comparison completions, annotated by *ERNIE4.0* for iterative DPO learning: (i) Generate 40K–80K weekly samples using baidu's internal data; (ii) Annotate 250K completions ($0.0174/K tokens); (iii) Train on 50K samples (1:5 quality ratio, $1K); (iiii) Complete in 1–2 months.

We evaluate algorithms by measuring their *Win rate* ($R_{\text{Win}}$ vs $R_{\text{Tie}}$ vs $R_{\text{Lose}}$) against a baseline policy, using sampling temperature 0.8 for generation and 0 for human-annotated ranking (scores 0–3). Proxies like *ERNIE-3.5* and *Ziya-LLaMA-7B-Reward* assess dialogue quality. A 'bad case' is defined as human-rated 0 for both answers. Win rates are computed from toxic comparison datasets with out-of-distribution model responses.

For better online performance evaluation, we split the *safe world view* dataset into two parts: 40k and 4k, using the 4k set for testing *margin reward* and *pairwise accuracy*. The *margin reward* is defined as the difference $r(\boldsymbol{x}, \boldsymbol{y}^w) - r(\boldsymbol{x}, \boldsymbol{y}^l)$ for preference triplets, while *pairwise accuracy* is defined as $r(\boldsymbol{x}, \boldsymbol{y}^w) > r(\boldsymbol{x}, \boldsymbol{y}^l)$.

## 6.3 Main Analysis

Shown as Table 1, our IOPO achieves the best performance (measured by win rate ($R_{\text{Win}}$ vs $R_{\text{Tie}}$ vs $R_{\text{Lose}}$) among all methods, particularly with ERNIE-3.5. A more comparison details analysis via *ERNIE-3.5* shown in appendix [4].

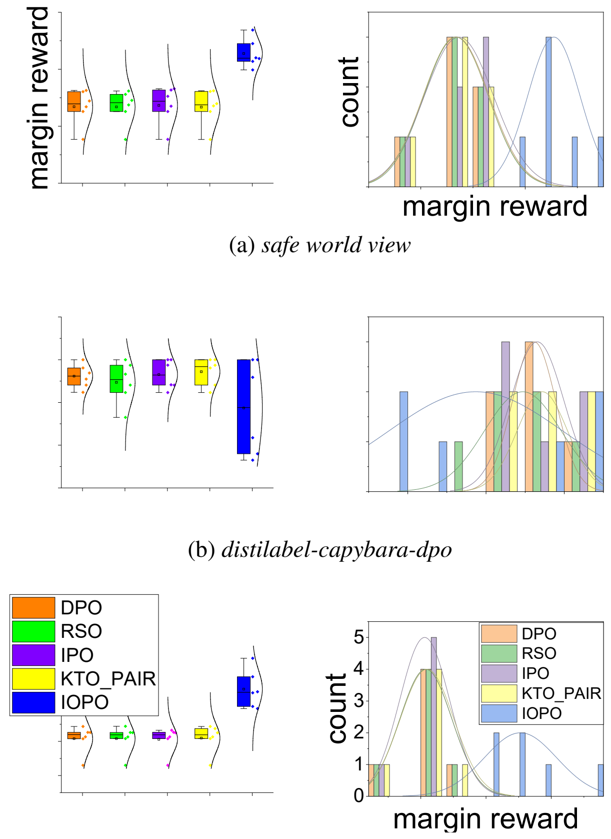
While RSO/HING outperform IOPO on *Ziya-7B* in *distilabel-capybara-dpo* and *orpo-dpo-mix-40k*, IOPO performs better on *ERNIE-3.5* due to: (1) Ziya's limited domain coverage (only 8.2% of its 40K training data comes from external sources), and (2) its smaller model size. **Rationale for**

[1] This dataset can be found on HuggingFace at https://hf-mirror.com/datasets/argilla/distilabel-capybara-dpo-7k-binarized

[2] https://hf-mirror.com/datasets/mlabonne/orpo-dpo-mix-40k

[3] *DPO*, *RSO*, *IPO*, and *KTO-PAIR* code is detailed on HuggingFace at https://hf-mirror.com/docs/trl/dpo_trainer.

[4] https://arxiv.org/submit/6429160/view



(a) *safe world view*

(b) *distilabel-capybara-dpo*

(c) *orpo-dpo-mix-40k*

Figure 3: Offline scenarios: margin reward distributions under a normal distribution (left) and count-based distribution (right).

**Baseline Retention show as:** Despite these limitations, we continue to use it as the baseline evaluator due to its exceptional ability to recognize toxic data and its official acknowledgment in GPT-4's benchmark.

A more detailed analysis is presented in Fig.3, distribution shift is reflected in normal distribution and distribution of *margin reward*, where it shows that the *margin reward* of training procedure for three datasets. For *distilabel-capybara-dpo*, depicted in Fig.3(b), the small scale of the dataset and the multi-domain (especially math) hinder the *margin reward* from increasing and stabilizing. For *orpo-dpo-mix-40k*, depicted in Fig.3(c) the *margin reward* increases compared to other methods but exhibits scatter. All in all, distribution in Table.3 with *win rate of IOPO vs. baseline* in Table.1, highlights the superiority of our IOPO method.

### 6.4 Ablation Analysis

We conduct an ablation evaluation with IOPO and Iter-DPO starting from the LLMs after DPO to isolate external factors. The SFT model is trained on *Baichuan2-7B-base*, and the DPO model builds on this SFT model. Human evaluations of 250 data pieces yield mean scores: SFT via *Baichuan2-7B-base* scores 2.156, DPO via *Baichuan2-7B-base* scores 2.164, SFT via *Baichuan2-7B-chat* scores 2.148, and DPO via *Baichuan2-7B-chat* also scores 2.148. To effectively implement the *Iter-DPO* algorithm and manage the high ranking costs of *ERNIE-4.0* (approximately 50k high-quality inference data), we select *Baichuan2-7B-base* as the base model, filtering 50k inference data points at a 1:5 ratio of high-quality to low-quality.

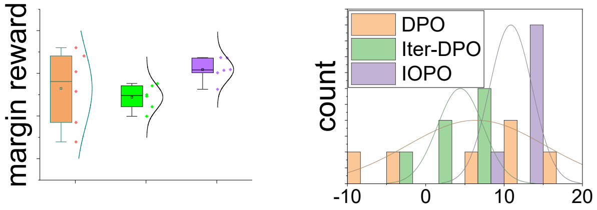


Figure 4: Online scenarios: margin reward distributions under normal distribution (left) and count-based distribution (right).

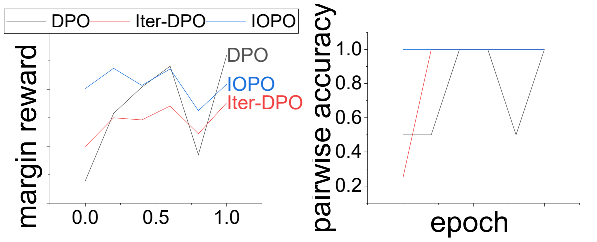


Figure 5: Online evaluation scenarios: margin reward (left) and pairwise accuracy (right) on the test set of *safe world view*. **Our method demonstrates more stable performance and superior results.**

As shown in Table.2, our method outperforms all other methods. While the DPO model excels in *margin reward*, it lags behind *Iter-DPO* and IOPO in *pairwise accuracy* during testing due to its poor generalization performance. Additionally, human evaluations highlight that *Iter-DPO* encounters *differences and modeling errors*, resulting in lower mean value scored by human than the DPO model but higher pairwise accuracy than DPO. In factual and safety evaluations, our IOPO achieves a better *win rate* based on human annotations and scored the highest mean value scored by human at a temperature of 0. The training procedure as depicted in Fig.4, DPO and Iter-DPO scatter more divergently in margin reward on *safe world view* than IOPO. As depicted in Fig.5, our method's performances on margin reward and pairwise accuracy on test set of *safe world view* much more better than others. All in all, our method perform the best among all the methods in online evaluations.

## 7 Conclusions

In conclusion, we propose the Indirect Online Preference Optimization (IOPO) algorithm, which mitigates distribution shifts and modeling errors while preserving linear complexity. Experiments on three real-world datasets show that IOPO surpasses state-of-the-art methods, reducing iteration time from months to one week and significantly lowering annotation costs—especially in online scenarios.

## Acknowledgments

This work is supported in part by National Key R&D Program of China under Grant Nos. 2022YFB3103700 and 2022YFB3103702, and National Natural Science Foundation of China under Grant Nos. 62272193, 62472194 and 62472196, and Jilin Science and Technology Research Project 20230101067JC.

## References

[Achiam *et al.*, 2023] 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. *arXiv preprint arXiv:2303.08774*, 2023.

[Azar *et al.*, 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 *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR, 2024.

[Bai *et al.*, 2022a] 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. *arXiv preprint arXiv:2204.05862*, 2022.

[Bai *et al.*, 2022b] 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*, 2022.

[Bos, 1980] Henk JM Bos. Newton, leibniz and the leibnizian tradition. *From the calculus to set theory*, pages 1630–1910, 1980.

[Bradley and Terry, 1952] 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.

[Cheng *et al.*, 2024] Pengyu Cheng, Yifan Yang, Jian Li, Yong Dai, Tianhao Hu, Peixin Cao, Nan Du, and Xiaolong Li. Adversarial preference optimization: Enhancing your alignment via rm-llm game. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 3705–3716, 2024.

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

[Ethayarajh *et al.*, 2024] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

[Haarnoja *et al.*, 2018] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[Kakade and Langford, 2002] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 267–274, 2002.

[Konda and Tsitsiklis, 1999] Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

[Kumar *et al.*, 2020] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

[Levine *et al.*, 2020] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[Littman, 1994] Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994.

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

[Ouyang *et al.*, 2022] 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. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[Pinto *et al.*, 2017] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International conference on machine learning*, pages 2817–2826. PMLR, 2017.

[Rafailov *et al.*, 2024] 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. *Advances in Neural Information Processing Systems*, 36, 2024.

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

[Tokdar and Kass, 2010] Surya T Tokdar and Robert E Kass. Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):54–60, 2010.

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

[Xiong *et al.*, 2024] Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. In *Forty-first International Conference on Machine Learning*, 2024.

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

[Yang *et al.*, 2023] Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*, 2023.

[Yuan *et al.*, 2024] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

[Zoph, 2016] B Zoph. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.