A Meta-Algorithm for Aligning LLMs with General Preferences

Anonymous Author(s)
Affiliation
Address
email

Abstract

Many alignment methods, including reinforcement learning from human feedback (RLHF), rely on the Bradley-Terry reward assumption, which is insufficient to capture the full range of general human preferences. To achieve robust alignment with general preferences, we model the alignment problem as a two-player zero-sum game, where the Nash equilibrium policy guarantees a 50% win rate against any competing policy. However, previous algorithms for finding the Nash policy either diverge or converge to a Nash policy in a modified game, even in a simple synthetic setting, thereby failing to maintain the 50% win rate guarantee against all other policies. We propose a meta-algorithm for language model alignment with general preferences, inspired by convergent algorithms in game theory. Theoretically, we prove that our meta-algorithm converges to an exact Nash policy. Additionally, our meta-algorithm is simple and can be integrated with many existing methods designed for RLHF and preference optimization with minimal changes. Experimental results demonstrate the effectiveness of the proposed framework when combined with existing preference policy optimization methods.

1 Introduction

Large Language Models (LLMs) [Brown et al., 2020, OpenAI, 2023, Dubey et al., 2024] have fundamentally transformed the fields of natural language processing and artificial intelligence. They excel in tasks ranging from text generation and translation to complex question answering and interactive dialogue systems. As these models become more integrated into daily life, a key challenge is ensuring they achieve high levels of alignment with human values and preferences.

One of the most widely adopted approaches to addressing this challenge is Reinforcement Learning from Human Feedback (RLHF) [Christiano et al., 2017, Ouyang et al., 2022]. This framework consists of two steps: first, learning a reward model from a dataset containing human preferences, and second, optimizing the LLM using the proximal policy optimization (PPO) algorithm [Schulman et al., 2017]. Recently, Rafailov et al. [2024] observed that the first step can be bypassed, proposing the direct preference optimization (DPO) algorithm, which directly optimizes the LLM from the dataset.

However, the aforementioned approaches crucially rely on the assumption that human preferences can be expressed using the Bradley-Terry (BT) model [Bradley and Terry, 1952]. Unfortunately, the BT model is too restrictive to capture the richness and complexity of human preferences. Specifically, the BT model can only induce *transitive* preferences—i.e., if more people favor A over B, and B over C, then more people must favor A over C. Such transitivity may not hold in the presence of diverse populations and is also incompatible with evidence from human decision-making [May, 1954, Tversky, 1969].

Submitted to Workshop on Fine-Tuning in Machine Learning (FITML@NeurIPS 2024). Do not distribute.

To overcome this limitation, recent research has begun to explore alignment under general preferences. Munos et al. [2024] formulate this alignment problem as a symmetric two-player zero-sum game, where both players' strategies are LLMs, and their payoffs are determined by the win rate against the opponent's LLM according to the preference model. The objective is to identify a Nash equilibrium policy that guarantees at least a 50% win rate against any other policy [Azar et al., 2024, Munos et al., 2024, Calandriello et al., 2024]. However, the trajectory of all the proposed algorithms either diverge or converge to the Nash policy of a modified game, thereby failing to maintain the 50% win rate guarantee against all other policies.

Our Contribution. We introduce a novel meta-algorithm, **Last-Iterate Nash Equilibrium Policy Optimization (LINE-PO)**, inspired by the proximal point method, a convergent algorithm for solving two-player zero-sum games [Nemirovski, 2004]. Our first observation is that many existing algorithms, including PPO [Schulman et al., 2017], DPO [Rafailov et al., 2024], IPO [Azar et al., 2024], SPPO [Wu et al., 2024], INPO [Zhang et al., 2024], etc., can be interpreted as implementations of the Prox operator [Nemirovski, 2004]. LINE-PO employs the Prox operator as its fundamental building block and provably *converges* to the Nash equilibrium policy in the *last iterate*, assuming the Prox operator can be computed exactly. This approach allows us to leverage many existing algorithms in a black-box manner. While several algorithms in the literature demonstrate average-iterate convergence to the Nash equilibrium policy, they all diverge in the last iterate. Unfortunately, iterate averaging can be cumbersome, particularly when deep-learning components are involved, as it may not be feasible to average the outputs of LLMs.[1] Compared to these algorithms, LINE-PO achieves the more desirable last-iterate convergence.

Additionally, we validate the effectiveness of LINE-PO in both synthetic and LLM settings.

Synthetic Setting. We construct a 3×3 two-player zero-sum preference game, and compare LINE-PO with a wide range of algorithms proposed in the literature. The result clearly shows that LINE-PO is the only algorithm that converges to the Nash equilibrium of the game in the last iterate.

LLM Setting. Furthermore, we evaluate the performance of LINE-PO against existing preference optimization algorithms under a real-world setting, where a pre-trained LLM, Qwen2-1.5B [Yang et al., 2024], is fine-tuned using different algorithms on the UltraFeedback [Cui et al., 2023] dataset, which is commonly used for alignment fine-tuning of LLMs. Our experimental results demonstrate the advantages of LINE-PO: it achieves at least 55% win rate compared against baseline algorithms including iterative algorithms such as iterative IPO [Azar et al., 2024] and INPO [Zhang et al., 2024].

2 Backgrouds

We use $\Delta(\mathcal{Z})$ to denote a distribution over a set $\mathcal{Z}$. We denote $x \in \mathcal{X}$ as an instruction where $\mathcal{X}$ is the instruction set. We assume a fixed distribution $\rho \in \Delta(\mathcal{X})$ over the instruction set. We denote $\mathcal{Y}$ as the response set and $y \in \mathcal{Y}$ as one response. Given any instruction $x \in \mathcal{X}$, an LLM policy π specifies the output distribution $\pi(\cdot \mid x) \in \Delta(\mathcal{Y})$. For distributions $p, q \in \Delta(\mathcal{Z})$, the Kullback-Leibler (KL) divergence is defined as $\mathrm{KL}(p||q) := \sum_{z \in \mathcal{Z}} p(z) \log \frac{p(z)}{q(z)}$. The sigmoid function is $\sigma(x) := \frac{e^x}{e^x+1}$. We use $\mathrm{supp}(p)$ to denote the support of a distribution p.

Preference Models In this paper, we focus on general preference models.

Definition 1 (General Preference Model)**.** *A general preference model $\mathbb{P} : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \to [0,1]$ satisfies $\mathbb{P}(y_1 \succ y_2 \mid x) = 1 - \mathbb{P}(y_2 \succ y_1 \mid x)$. When we query $\mathbb{P}$ with (x, y_1, y_2), it outputs 1 with probability $\mathbb{P}(y_1 \succ y_2 \mid x)$ meaning y_1 is preferred over y_2, and it outputs 0 otherwise.*

We define $\mathbb{P}(\pi_1 \succ \pi_2) := \mathbb{E}_{x \sim \rho}[\mathbb{E}_{y_1 \sim \pi_1, y_2 \sim \pi_2}[\mathbb{P}(y_1 \succ y_2 \mid x)]]$ as the *win rate* of π_1 over π_2 under preference model $\mathbb{P}$. A special case of the general preference model is the Bradley-Terry (BT) model, which assumes a reward function parameterizes the preference. We review alignment under the BT model in Appendix A.

[1] Storing all LLMs produced during training could solve this, but it is highly space-inefficient and, to our knowledge, has not been implemented.

Table 1: Property comparison of different preference optimization algorithms. (*) Means convergence in the original game $J(\pi_1, \pi_2)$

Algorithm	General Preference	Regularized Game Solver	Last-Iterate Convergence*
DPO [Rafailov et al., 2024]	✗	✗	✗
IPO [Azar et al., 2024]	✓	✗	✗
SPPO [Wu et al., 2024]	✓	✗	✗
INPO [Zhang et al., 2024]	✓	✓	✗
LINE-PO	✓	✓	✓

Definition 2 (Bradley-Terry Model). *A preference model* $\mathbb{P}$ *satisfies the* Bradley-Terry (BT) assumption *if there exists a reward function* $r^* : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ *such that*

$$\mathbb{P}(y_1 \succ y_2 \mid x) = \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)} = \sigma(r^*(x, y_1) - r^*(x, y_2)).$$

2.1 Alignment with General Preference Models

The BT model assumption is insufficient to capture the full range of general human preferences [Munos et al., 2024, Swamy et al., 2024]. To achieve robust alignment with general preferences, we model the policy optimization problem as a two-player zero-sum game with the objective function as follows:[2]

$$J(\pi_1, \pi_2) := \mathbb{P}(\pi_1 \succ \pi_2) - \frac{1}{2} = \mathbb{E}_{x \sim \rho}[\mathbb{E}_{y_1 \sim \pi_1, y_2 \sim \pi_2}[\mathbb{P}(y_1 \succ y_2 \mid x)]] - \frac{1}{2}. \tag{1}$$

In this game, the max-player controls π_1 and tries to maximize $J(\pi_1, \pi_2)$ while the min-player controls π_2 and tries to minimize $J(\pi_1, \pi_2)$. We focus only on policies with $\Pi := \{\pi : \mathrm{supp}(\pi) \subseteq \mathrm{supp}(\pi_{\mathrm{sft}})\}$ in the support of the initial SFT policy. A Nash equilibrium policy $(\pi_1^\star, \pi_2^\star)$ satisfies

$$\pi_1^\star, \pi_2^\star \in \operatorname*{argmax}_{\pi_1 \in \Pi} \operatorname*{argmin}_{\pi_2 \in \Pi} J(\pi_1, \pi_2), \quad J(\pi_1, \pi_2^\star) \le J(\pi_1^\star, \pi_2^\star) \le J(\pi_1^\star, \pi_2), \forall \pi_1, \pi_2 \in \Pi.$$

Since $J(\pi_1, \pi_2)$ is symmetric, the game has a symmetric Nash equilibrium $(\pi^\star, \pi^\star)$. Moreover, the Nash equilibrium policy $\pi^\star$ guarantees that for any other policy π, its win rate is at least $\mathbb{P}(\pi^\star \succ \pi) \ge \mathbb{P}(\pi^\star \succ \pi^\star) = 50\%$. We call this property *robust alignment*. Our goal is to find a policy with robust alignment.

Existing online iterative preference optimization methods designed for or applicable to the original game including iterative IPO [Azar et al., 2024] and SPPO [Wu et al., 2024], are based on Multiplicative Weights Update, and thus *diverge* as we show in Section 4. There is also a line of works including Nash-MD [Munos et al., 2024, Ye et al., 2024], Online IPO [Calandriello et al., 2024], INPO [Zhang et al., 2024] aim to find the Nash equilibrium of a modified KL-regularized game:

$$J_\tau(\pi_1, \pi_2, \pi_{\mathrm{ref}}) := J(\pi_1, \pi_2) - \tau \mathbb{E}_{x \sim \rho}[\mathrm{KL}(\pi_1(\cdot \mid x) || \pi_{\mathrm{ref}}(\cdot \mid x))] + \tau \mathbb{E}_{x \sim \rho}[\mathrm{KL}(\pi_2(\cdot \mid x) || \pi_{\mathrm{ref}}(\cdot \mid x))].$$

The additional KL regularization terms in the objective are introduced for training stability. However, the Nash equilibrium of the modified game no longer achieves robust alignment, i.e., has a win rate of at least 50% against any competing policy.

Moreover, most existing theoretical convergence guarantees only hold for the average iterate, i.e., the uniform mixture of training iterates, which is not used in practice. We focus on designing algorithms with provable last-iterate convergence to Nash equilibrium, which aligns with practice and is more space-efficient [Munos et al., 2024].

In the next section, we propose a meta-algorithm that uses algorithms designed for the regularized game $J_\tau(\pi_1, \pi_2, \pi_{\mathrm{ref}})$ or other preference optimization methods as black-boxes to find Nash equilibrium of $J(\pi_1, \pi_2)$ (1), thereby achieving robust alignment.

3 Last-Iterate Nash Equilibrium Policy Optimization

We propose an extremely simple meta-algorithm, Last-Iterate Nash Equilibrium Policy Optimization (LINE-PO, Algorithm 1), for robustly aligning LLMs with general preferences. LINE-PO is an

[2] We introduce the constant $\frac{1}{2}$ only to ensure the game is zero-sum and it has no effect on its Nash equilibria.

Algorithm 1: Last-Iterate Convergent Nash Equilibrium Policy Optimization (LINE-PO)

Input: Initial policy π_{sft}, preference oracle $\mathbb{P}$, regularization $\tau > 0$

Initialize $\pi^1, \pi_{\text{ref}} \leftarrow \pi_{\text{sft}}$

for $t = 1, 2, \ldots, T-1$ **do**

  $\pi^{t+1} \leftarrow \text{argmax}_{\pi_1} \min_{\pi_2} J_\tau(\pi_1, \pi_2, \pi_{\text{ref}})$ using Algorithm 2

  $\pi_{\text{ref}} \leftarrow \pi^{t+1}$

return π^T

online iterative algorithm inspired by the classic Conceptual Prox method [Nemirovski, 2004] first introduced in the optimization theory community. This method has recently been applied to finding a Nash equilibrium in zero-sum games [Perolat et al., 2021, Abe et al., 2024] and has had notable success in training advanced game AI models [Perolat et al., 2022].

3.1 LINE-PO

In each iteration t, LINE-PO updates the next-iteration policy π^{t+1} as the Nash equilibrium policy of a regularized game $J_\tau(\pi_1, \pi_2, \pi_{\text{ref}})$ using the current policy as reference $\pi_{\text{ref}} = \pi^t$. The rationale behind LINE-PO is simple: update the reference policy when there is no improvement in the regularized game. Denote $\pi^\star$ the Nash equilibrium of the original game. We show that KL divergence to $\pi^\star$ is monotonically decreasing: $\text{KL}(\pi^\star || \pi^{t+1}) \le \text{KL}(\pi^\star || \pi^t)$. Since π^{t+1} is closer to the Nash equilibrium than π^t, LINE-PO updates the reference policy from π^t to π^{t+1} for further optimization. We also remark that in LINE-PO, the regularization amount $\tau > 0$ does not need to decrease and could be kept constant.

Each iteration of LINE-PO requires solving a zero-sum game with additional KL regularization $J_\tau(\pi_1, \pi_2, \pi_{\text{ref}})$. We will show momentarily that many existing policy optimization methods for alignment can be applied to the KL regularized game and have exponentially fast convergence. We prove the meta-algorithm LINE-PO achieves last-iterate convergence to a Nash equilibrium with robust alignment property, which appears to be the first in the context of LLM alignment.

Theorem 1. *We assume that there exists a Nash equilibrium $\pi^\star$ of $J(\pi_1, \pi_2)$ (defined in (1)) such that $\text{supp}(\pi^\star) = \text{supp}(\pi_{\text{sft}})$. In every iteration $t \ge 1$, it holds that $\text{KL}(\pi^\star || \pi^{t+1}) \le \text{KL}(\pi^\star || \pi^t)$. Moreover, LINE-PO has last-iterate convergence, i.e., $\lim_{t \to \infty} \pi^t$ exists and is a Nash equilibrium.*

3.2 Solving a Regularized Game

We show how to solve the Nash equilibrium of the regularized game $J_\tau(\pi_1, \pi_2, \pi_{\text{ref}})$ using the Mirror Descent (MD) algorithm and how to implement MD using existing policy optimization algorithms. For simplicity, we consider policy $\pi \in \Delta(\mathcal{Y})$ and omit the dependence on the instruction x. All discussions can be extended to the contextual setting in a straightforward way.

Mirror Descent and Multiplicative Weights Update Mirror Descent (MD) is a classical family of optimization algorithms. An important member of this family is the Multiplicative Weights Update (MWU) algorithm, which is MD with negative entropy regularization. For a maximization problem $\max_\pi f(\pi)$, given an existing policy π^t, MWU computes the update π^{t+1} as follows:

$$\pi^{t+1} := \underset{\pi}{\text{argmax}} \left\langle \nabla f(\pi^t), \pi \right\rangle - \eta^{-1} \cdot \text{KL}(\pi || \pi^t). \tag{2}$$

Note that RLHF in (4) is equivalent to one step of MWU if we interpret the reward r as the gradient $\nabla f(\pi_{\text{ref}})$.

Prox operator. The update rule of MWU can be compactly written using the *prox operator* as shown in Algorithm 2.[3] Fix a 1-strongly convex function $\varphi : \mathcal{Z} \to \mathbb{R}$ over a closed convex set $\mathcal{Z} \subset \mathbb{R}^n$. The *Bregman divergence* induced by φ is

$$D_\varphi(\cdot || \cdot) : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}_{\ge 0},$$
$$D_\varphi(z || z') := \varphi(z) - \varphi(z') - \langle \nabla \varphi(z'), z - z' \rangle.$$

[3] The prox operator is also called the prox-mapping [Nemirovski, 2004].

Algorithm 2: Regularized game solver

Input: Reference policy π_{ref}, preference oracle $\mathbb{P}$, regularization $\tau > 0$, step size $\eta > 0$, number of iterations $K \geq 1$

Output: A Nash policy $\text{argmax}_{\pi^1} \min_{\pi_2} J_\tau(\pi_1, \pi_2, \pi_{\text{ref}})$

Initialize $\mu^1 \leftarrow \pi_{\text{ref}}$

for $k = 1, 2, \ldots, K-1$ **do**

  $g_\tau^k \leftarrow \nabla_\mu(\mathbb{P}(\mu \succ \mu_k) - \tau \, \text{KL}(\mu || \pi_{\text{ref}})) = \mathbb{P}(\cdot \succ \mu_k) - \tau(\log \frac{\mu_k(\cdot)}{\pi_{\text{ref}}(\cdot)} + 1)$

  $\mu^{k+1} \leftarrow \text{Prox}(\mu_k, \eta g_\tau^k)$

return μ_K

Given a reference point $z \in \mathcal{Z}$ and a vector $g \in \mathbb{R}^n$, the prox operator $\text{Prox}(z, g)$ generalizes the notion of a gradient ascent step from z in the direction of g.

Definition 3 (Prox Operator)**.** *For a strongly convex regularizer φ, the prox operator is defined as*

$$\text{Prox}(z, g) := \underset{z'}{\text{argmax}} \, \langle g, z' \rangle - D_\varphi(z' || z) = \underset{z'}{\text{argmax}} \, \langle g + \nabla \varphi(z), z' \rangle - \varphi(z'). \tag{3}$$

When $\varphi(z) = \frac{1}{2}\|z\|_2^2$ is the ℓ_2 regularizer, the prox operator $\text{Prox}(z, g) = \Pi_{\mathcal{Z}}[z + g]$ is the exactly the projected gradient ascent step. In this paper, without additional notes, we choose $\varphi = \sum_{i=1}^n z[i] \ln z[i]$ as the negative entropy regularizer and the corresponding Bregman divergence D_φ is the KL divergence.

The flexibility of the prox operator lies in the choice of g for different objectives. In RLHF, g is the reward model r and we compute the optimal policy $\pi^\star = \text{Prox}(\pi_{\text{ref}}, \eta r)$. For vanilla MWU, g is the gradient $\nabla f(\pi^t)$ and we update $\pi^{t+1} = \text{Prox}(\pi^t, \eta \nabla f(\pi^t))$. When a preference model $\mathbb{P}$ is available, we can choose g as the preference function $\mathbb{P}(\cdot \succ \pi)$ over the current policy π. For our theoretical results, we assume the prox operator Prox can be evaluated exactly or approximately. Practically, we can use many existing preference optimization methods to compute the prox operator as shown in the next section.

Exponentially Fast Convergence Denote $\pi_\tau^\star$ the Nash equilibrium of the KL regularized game $J_\tau(\pi_1, \pi_2, \pi_{\text{ref}})$, which is τ-strongly monotone. We can apply classical results to show that MWU (Algorithm 2) achieves linear last-iterate convergence rate: the distance to the Nash equilibrium $\pi_\tau^\star$ decreases exponentially fast.

Theorem 2. *For appropriate step size $\eta > 0$, Algorithm 2 guarantees for every $k \geq 1$,* $\text{KL}(\pi_\tau^\star || \mu^{k+1}) \leq (1 - \frac{\eta\tau}{2})^k \, \text{KL}(\pi_\tau^\star || \pi_{\text{ref}})$.

3.3 Computing the prox operator

We show how to compute the prox operator in practical large-scale applications like LLM alignment. Specifically, we show that many existing algorithms designed for RLHF and preference optimization with neural network parameters can be adapted to solve the prox operator $\text{Prox}(\pi, \eta g)$ ($\eta > 0$ is the step size). These algorithms include RL algorithms like PPO, and loss-minimization algorithms like DPO, IPO, SPPO, DRO, each of which may be preferred in certain settings. Our contribution here is not proposing new algorithms but unifying existing diverse preference methods through the perspective of computing the prox operator. Due to space limit, we defer the discussion to Appendix F. This perspective opens the possibility of applying other algorithms from online learning and optimization to robust LLM alignment and we include implementation for two other algorithms in Appendix H.

4 Synthetic Experiments

We conduct experiments on a simple bandit problem with $\mathcal{Y} = \{y_a, y_b, y_c\}$ and non-BT preference model over $\mathcal{Y}$. Specifically, we set $\mathbb{P}[y_b \succ y_a] = \mathbb{P}[y_c \succ y_b] = 0.9$ and $\mathbb{P}[y_a \succ y_c] = 0.8$. We can observe that the preference is intransitive and exhibits a preference cycle $y_c \succ y_b \succ y_a \succ y_c$.

Experiments using noiseless gradient We present numerical results of mirror-descent (MD) algorithms (equivalent to MWU) and LINE-PO (Algorithm 1) in Figure 1. We can see that the MD algorithm diverges from the unique Nash equilibrium and suffers a large equilibrium gap, while LINE-PO achieves fast last-iterate convergence to the Nash equilibrium, aligned with our theoretical results (Theorem 1).

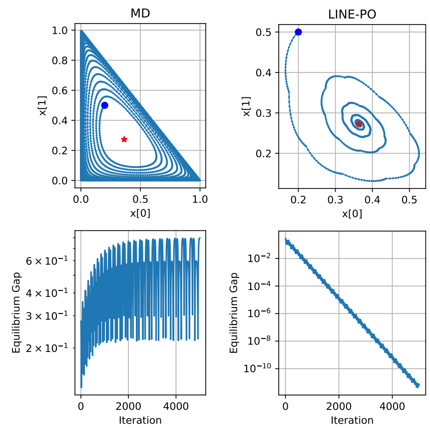

Figure 1: Dyanmics on a simple 3-dimensional preference game. The unique Nash equilibrium is $[4/11, 3/11, 3/11]$ represented as red star. We initialize all algorithms at the blue dot point $[0.2, 0.5, 0.3]$.

Experiments using preference samples Since the popular iterative DPO algorithm does not contain a gradient step, we also conduct experiments with only Oracle query access to the preference model. We compare the performance of various algorithms, including iterative DPO, iterative IPO, SPPO, and INPO and present results in Figure 2. We remark that iterative DPO and iterative IPO both diverge in the last iterate; INPO converges to a point that is not Nash equilibrium and does not guarantee robust alignment; LINE-PO is the only algorithm that achieves last-iterate convergence to the Nash equilibrium. We defer a more detailed discussion to Appendix I.

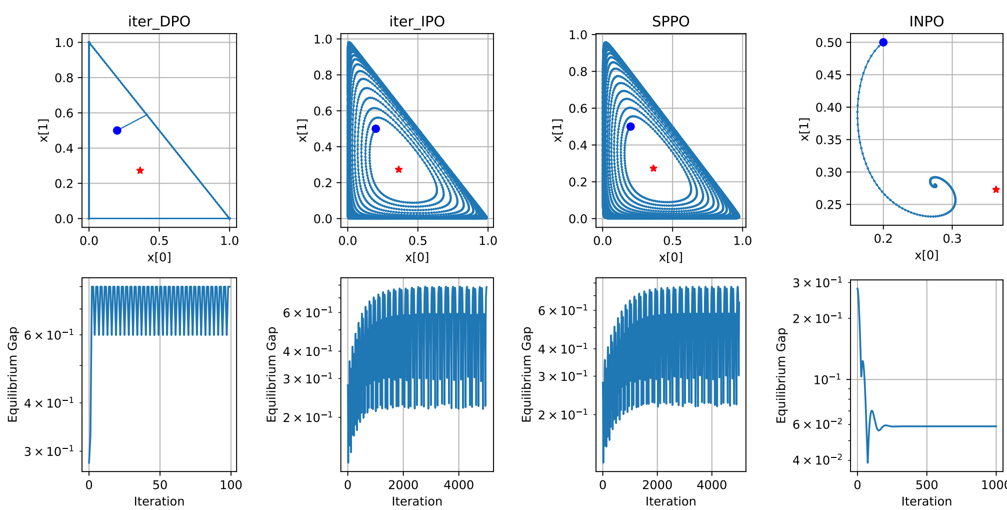

Figure 2: Dyanmics on a simple 3-dimensional preference game. The unique Nash equilibrium is $[4/11, 3/11, 3/11]$ represented as red star. We initialize all algorithms at the blue dot point $[0.2, 0.5, 0.3]$.

5 Real-World Experiments

Apart from the controlled synthetic experiments, we conduct experiments with a pre-trained LLM, Qwen2-1.5B [Yang et al., 2024], on a commonly used dataset UltraFeedback [Cui et al., 2023] to show the effectiveness of LINE-PO under the real-world preference optimization setting.

5.1 Experimental Settings

Datasets We use the UltraFeedback dataset, specifically its binarized version for preference fine-tuning.[4] It contains 64K data examples consisting of a user instruction and a positive-negative output pair annotated by GPT-4. The instructions contained in this dataset cover a wide range of instruction types, making it suitable to study preference optimization in a real-world setting. Since we focus on online and iterative preference optimization, only the instructions are used because the output pairs will be generated and annotated online. In addition, to reduce the computation cost, the instructions are randomly split into 6 equal-size subsets. Each subset therefore contains around 10K instructions and is used in one training iteration.

Preference Oracle The preference oracle we chose is Llama-3-OffsetBias-8B [Park et al., 2024a], which is a pairwise preference model that predicts which output is better given an instruction and an output pair. Fine-tuned from Meta-Llama-3-8B-Instruct [Dubey et al., 2024], it achieves strong performance on various human preference alignment benchmarks on RewardBench [Lambert et al., 2024]. We chose it as the preference oracle since it strikes a balance between computation efficiency and alignment with human preferences, making it suitable for iterative preference optimization.

Online Preference Data Generation To construct the preference data, i.e., output pairs with a preference annotation specifying which one is better, we adopt the setting of Zhang et al. [2024] by sampling 5 candidate outputs for each instruction with a temperature of 0.8 and applying the preference oracle to compare all the output pairs constructed. The best and the worst candidate outputs, derived from the pairwise comparison results, are then selected to form a data point.

Baselines We include the following baselines for comparisons with LINE-PO: (1) SFT, which fine-tunes the pre-trained Qwen2-1.5B on the UltraChat dataset, and the resulted checkpoint serves as the start point and/or the reference policy for the other training algorithms; (2) vanilla online DPO [Rafailov et al., 2024] and (3) vanilla online IPO [Azar et al., 2024], where one training iteration is performed over the entire instruction set of UltraFeedback; (4) INPO [Zhang et al., 2024], where at each iteration the training is performed on one data split; (5) iterative IPO, which has a similar training setting to INPO but without the KL constraint from the reference policy.

Evaluations We use the instructions in a widely used benchmark, AlpacaEval [Li et al., 2023], to construct the test set, since these instructions are diverse and cover various task scenarios. However, we choose not to use the default evaluator of the AlpacaEval benchmark, GPT-4, to perform the evaluation, but instead use the same preference oracle used in data generation, Llama-3-OffsetBias-8B, as the evaluator. This is because we aim for a controlled experimental setting – the preference oracle that the model learns to fit should also be the one used to evaluate the model performance.

Training Details We follow the training recipe proposed in Tunstall et al. [2023] for the experiments. Specifically, at each training iteration, the models are fine-tuned for 3 epochs with the batch size setting to 32 and with a linear learning rate scheduler. The checkpoints are selected based on their validation loss on the UltraFeedback dataset. As for the hyper-parameters, we perform a grid search for the strength of the KL regularization, η^{-1}, in vanilla DPO and IPO. Specifically, we found that DPO achieves the best performance when it is set to 0.01, while IPO achieves the best performance when η^{-1} is set within the range of 0.01 - 0.002. We then choose the value of η to be 0.002 to encourage larger learning steps. This value of η is also used for iterative IPO and INPO. INPO has another hyper-parameter τ which controls the strength of the KL regularization from the reference policy. We determine its value following the setting of Zhang et al. [2024], where $\eta\tau$ is set to a fixed ratio, $1/3$. Regarding LINE-PO, the second training round starts when the first training round based on INPO begins to converge/overfit, and η^{-1} is set to 0.01 for the second round for training stability.

5.2 Result Analysis

Figure 3 presents the training dynamics of three iterative preference optimization algorithms we compared: iterative IPO (Iter-IPO), INPO, and LINE-PO, which are demonstrated by their checkpoints' win rates against the SFT checkpoint and the average length of their outputs. For INPO and LINE-PO, the model is trained for up to 18 iterations, which are equivalent to 3 training rounds over the entire instruction set since it has been split into 6 subsets. We note that:

[4] https://huggingface.co/datasets/HuggingFaceH4/ultrafeedback_binarized.

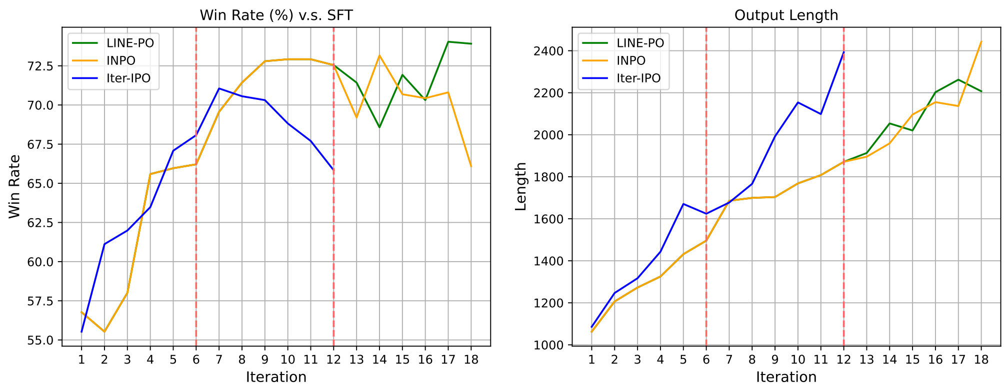

Figure 3: Comparisons of Iterative IPO (Iter-IPO), INPO, and LINE-PO. The win rate of the trained checkpoints against the SFT checkpoint, and the average length of the outputs are compared. The red vertical lines mark the end of one training round – a complete iteration over the 6 data splits.

Table 2: Performance comparison of different training algorithms. The row v.s. column win rate (%) is reported. For INPO, we report its performance with 2-round (R2) and 3-round (R3) training.

Row/Col	SFT	DPO	IPO	Iter-IPO	INPO-R2	INPO-R3	LINE-PO	Avg
Iter-IPO	65.84	56.40	54.04	50.00	47.83	46.21	39.01	51.33
INPO-R2	72.55	60.25	58.39	52.17	50.00	49.32	41.37	54.86
INPO-R3	66.09	60.25	58.51	53.79	50.68	50.00	44.97	54.90
LINE-PO	**73.91**	**66.71**	**66.21**	**60.99**	**58.63**	**55.03**	**50.00**	**61.64**

(1) Iter-IPO shows a quicker improvement rate at the beginning of the training, but its performance against the SFT checkpoint starts to degrade in the second training round, which indicates the inherent instability of this training algorithm.

(2) INPO archives a relatively stable win rate against SFT at the end of the second training round. However, its win rate starts to slightly degrade in the third training round. We suspect this suggests that INPO has started to converge and/or overfit. Therefore, for LINE-PO, which shares the same training trajectory as INPO for the first two training rounds, we update the reference policy at the beginning of the third training round, following the optimization process described in Algorithm 1.

(3) LINE-PO is able to further improve the model performance with the updated reference policy. Notably, it also results in the shortest outputs compared to Iter-IPO and INPO, suggesting that it is more robust to the length bias of the preference models which preference optimization algorithms tend to exploit [Park et al., 2024b].

Table 2 provides pairwise comparisons between the final checkpoints of the iterative preference optimization algorithms and a few baselines. It demonstrates the clear advantage of LINE-PO, which is able to achieve an above 50% win rate against all the other checkpoints. In contrast, Iter-IPO can only outperform the vanilla DPO and IPO settings. Regarding INPO, we found that the average win rate of its checkpoint after the third training round (INPO-R3) is only slightly higher than that of its intermediate checkpoint at the end of the second training round (INPO-R2) (54.90 vs. 54.86), suggesting that its performance plateaued by the end of the second training round.

6 Conclusion

We have proposed LINE-PO, a meta-algorithm for preference optimization that provably converges to the Nash equilibrium policy in the last iterate. We have provided a theoretical analysis of the properties of LINE-PO and have empirically demonstrated its effectiveness under both synthetic and real-world experimental settings. We believe LINE-PO has significant potential to enhance the performance of LLMs in the alignment fine-tuning setting, due to its theoretical guarantees and flexibility, as it can be integrated with existing learning algorithms while overcoming their limitations.

References

Kenshi Abe, Kaito Ariu, Mitsuki Sakamoto, and Atsushi Iwasaki. Adaptively perturbed mirror descent for learning in games. In *Proceedings of the 41st International Conference on Machine Learning*, 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.

Heinz H. Bauschke, Walaa M. Moursi, and Xianfu Wang. Generalized monotone operators and their averaged resolvents. *Math. Program.*, 189(1):55–74, 2021. doi: 10.1007/S10107-020-01500-6. URL https://doi.org/10.1007/s10107-020-01500-6.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324, 1952. URL https://api.semanticscholar.org/CorpusID:125209808.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Yang Cai, Argyris Oikonomou, and Weiqiang Zheng. Finite-time last-iterate convergence for learning in multi-player games. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Daniele Calandriello, Zhaohan Daniel Guo, Remi Munos, Mark Rowland, Yunhao Tang, Bernardo Avila Pires, Pierre Harvey Richemond, Charline Le Lan, Michal Valko, Tianqi Liu, Rishabh Joshi, Zeyu Zheng, and Bilal Piot. Human alignment of large language models through online preference optimisation. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=2RQqg2Y7Y6.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4299–4307, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html.

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. *arXiv preprint arXiv:2310.01377*, 2023.

Constantinos Daskalakis and Ioannis Panageas. The limit points of (optimistic) gradient descent in min-max optimization. In *the 32nd Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David

Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle

Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. KTO: model alignment as prospect theoretic optimization. *CoRR*, abs/2402.01306, 2024. doi: 10.48550/ARXIV.2402.01306. URL https://doi.org/10.48550/arXiv.2402.01306.

Francisco Facchinei and Jong-Shi Pang. *Finite-dimensional variational inequalities and complementarity problems*. Springer, 2003.

Zhaolin Gao, Jonathan D. Chang, Wenhao Zhan, Owen Oertell, Gokul Swamy, Kianté Brantley, Thorsten Joachims, J. Andrew Bagnell, Jason D. Lee, and Wen Sun. REBEL: reinforcement learning via regressing relative rewards. *CoRR*, abs/2404.16767, 2024. doi: 10.48550/ARXIV.2404.16767. URL https://doi.org/10.48550/arXiv.2404.16767.

Noah Golowich, Sarath Pattathil, and Constantinos Daskalakis. Tight last-iterate convergence rates for no-regret learning in multi-player games. *Advances in neural information processing systems (NeurIPS)*, 2020a.

Noah Golowich, Sarath Pattathil, Constantinos Daskalakis, and Asuman Ozdaglar. Last iterate is slower than averaged iterate in smooth convex-concave saddle point problems. In *Conference on Learning Theory (COLT)*, 2020b.

Eduard Gorbunov, Adrien B. Taylor, and Gauthier Gidel. Last-iterate convergence of optimistic gradient method for monotone variational inequalities. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/893cd874ba98afa54ae9e385a24a83ac-Abstract-Conference.html.

Yu-Guan Hsieh, Kimon Antonakopoulos, and Panayotis Mertikopoulos. Adaptive learning in continuous games: Optimal regret bounds and convergence to nash equilibrium. In *Conference on Learning Theory*, pages 2388–2422. PMLR, 2021.

A. N. Iusem, T. Pennanen., and B. F. Svaiter. Inexact variants of the proximal point algorithm without monotonicity. *SIAM Journal on Optimization*, 13(4):1080–1097, 2003. doi: 10.1137/S1052623401399587. URL https://doi.org/10.1137/S1052623401399587.

G. M. Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976. URL https://ci.nii.ac.jp/naid/10017556617/.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*, 2024.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 5 2023.

Kenneth O May. Intransitivity, utility, and the aggregation of preference patterns. *Econometrica: Journal of the Econometric Society*, pages 1–13, 1954.

Panayotis Mertikopoulos, Christos H. Papadimitriou, and Georgios Piliouras. Cycles in adversarial regularized learning. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2703–2717. SIAM, 2018. doi: 10.1137/1.9781611975031.172. URL https://doi.org/10.1137/1.9781611975031.172.

Aryan Mokhtari, Asuman Ozdaglar, and Sarath Pattathil. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020a.

Aryan Mokhtari, Asuman E Ozdaglar, and Sarath Pattathil. Convergence rate of $\mathcal{O}(1/k)$ for optimistic gradient and extragradient methods in smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 30(4):3230–3251, 2020b.

Remi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Côme Fiegel, et al. Nash learning from human feedback. In *Forty-first International Conference on Machine Learning*, 2024.

Arkadi Nemirovski. Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774. URL https://doi.org/10.48550/arXiv.2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Junsoo Park, Seungyeon Jwa, Meiying Ren, Daeyoung Kim, and Sanghyuk Choi. Offsetbias: Leveraging debiased data for tuning evaluators. *arXiv preprint arXiv:2407.06551*, 2024a.

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

Julien Perolat, Remi Munos, Jean-Baptiste Lespiau, Shayegan Omidshafiei, Mark Rowland, Pedro Ortega, Neil Burch, Thomas Anthony, David Balduzzi, Bart De Vylder, et al. From poincaré recurrence to convergence in imperfect information games: Finding equilibrium via regularization. In *International Conference on Machine Learning*, pages 8525–8535. PMLR, 2021.

Julien Perolat, Bart De Vylder, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer, Paul Muller, Jerome T Connor, Neil Burch, Thomas Anthony, et al. Mastering the game of stratego with model-free multiagent reinforcement learning. *Science*, 378(6623):990–996, 2022.

Leonid Denisovich Popov. A modification of the Arrow-Hurwicz method for search of saddle points. *Mathematical notes of the Academy of Sciences of the USSR*, 28(5):845–848, 1980. Publisher: Springer.

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.

Sasha Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. *Advances in Neural Information Processing Systems*, 2013.

Pierre Harvey Richemond, Yunhao Tang, Daniel Guo, Daniele Calandriello, Mohammad Gheshlaghi Azar, Rafael Rafailov, Bernardo Avila Pires, Eugene Tarassov, Lucas Spangher, Will Ellsworth, et al. Offline regularised reinforcement learning for large language models alignment. *arXiv preprint arXiv:2405.19107*, 2024.

R. Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976. doi: 10.1137/0314056. URL https://doi.org/10.1137/0314056.

Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacroce, Ahmed Awadallah, and Tengyang Xie. Direct nash optimization: Teaching language models to self-improve with general preferences. *CoRR*, abs/2404.03715, 2024. doi: 10.48550/ARXIV.2404.03715. URL https://doi.org/10.48550/arXiv.2404.03715.

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

Gokul Swamy, Christoph Dann, Rahul Kidambi, Zhiwei Steven Wu, and Alekh Agarwal. A minimaximalist approach to reinforcement learning from human feedback, 2024.

Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E Schapire. Fast convergence of regularized learning in games. *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Remi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Avila Pires, and Bilal Piot. Generalized preference optimization: A unified approach to offline alignment. In *Forty-first International Conference on Machine Learning*, 2024.

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. *arXiv preprint arXiv:2310.16944*, 2023.

Amos Tversky. Intransitivity of preferences. *Psychological review*, 76(1):31, 1969.

Chen-Yu Wei, Chung-Wei Lee, Mengxiao Zhang, and Haipeng Luo. Linear last-iterate convergence in constrained saddle-point optimization. In *International Conference on Learning Representations (ICLR)*, 2021.

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.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Chenlu Ye, Wei Xiong, Yuheng Zhang, Nan Jiang, and Tong Zhang. A theoretical analysis of nash learning from human feedback under general kl-regularized preference. *arXiv preprint arXiv:2402.07314*, 2024.

Yuheng Zhang, Dian Yu, Baolin Peng, Linfeng Song, Ye Tian, Mingyue Huo, Nan Jiang, Haitao Mi, and Dong Yu. Iterative nash policy optimization: Aligning llms with general preferences via no-regret learning. *arXiv preprint arXiv:2407.00617*, 2024.

Contents

1 Introduction **1**

2 Backgrouds **2**

2.1 Alignment with General Preference Models 3

3 Last-Iterate Nash Equilibrium Policy Optimization **3**

3.1 LINE-PO 4

3.2 Solving a Regularized Game 4

3.3 Computing the prox operator 5

4 Synthetic Experiments **5**

5 Real-World Experiments **6**

5.1 Experimental Settings 7

5.2 Result Analysis 7

6 Conclusion **8**

A Alignment under the Bradley-Terry Model Assumption **14**

B Related Work **15**

C Properties of the Prox Operator **16**

D Proof of Theorem 1 **16**

E Proof of Theorem 2 **18**

F Computing the Prox Operator using Preference Learning Methods **18**

G Practical Implementation of Algorithms **20**

H Implementation of Mirror-Prox and Optimistic Multiplicative Weights Update **20**

I Detailed Discussion on Synthetic Experiments **21**

A Alignment under the Bradley-Terry Model Assumption

RLHF The canonical formulation of Reinforcement Learning from Human Feedback (RLHF) is to first learn a reward function r under the BT model and then find the optimal KL regularized policy π^* with respect to the learned reward function r:

$$\pi^* := \arg\max_{\pi} \mathbb{E}_{x\sim\rho, y\sim\pi(\cdot|x)}\big[r(x,y) - \eta^{-1}\,\mathrm{KL}(\pi(\cdot \mid x)||\pi_{\mathrm{ref}}(\cdot \mid x))\big], \tag{4}$$

where $\eta^{-1} > 0$ controls the regularization, and π_{ref} is the initial reference model, usually the policy π_{sft} obtained from pre-training and supervised fine-tuning.

DPO Rafailov et al. [2024] observe that the regularized optimization problem (4) has a closed-form solution : for any x and y,

$$\pi^*(y \mid x) = \frac{\pi_{\text{ref}}(y \mid x) \exp\left(\eta r(x, y)\right)}{Z_x}, \tag{5}$$

where $Z_x = \mathbb{E}_{y \sim \pi_{\text{ref}}(\cdot|x)}[\exp(\frac{1}{\eta} r(y, x))]$ is the normalization constant known as the partition function. In (5), we see that π^* implicitly parameterizes the reward function r. Rafailov et al. [2024] propose direct preference optimization (DPO) to learn the optimal policy using the maximum likelihood objective directly:

$$\ell_{\text{DPO}}(\pi; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}}\left[\log \sigma\left(\eta^{-1} \log \frac{\pi(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \eta^{-1} \log \frac{\pi(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right],$$

where $\mathcal{D}$ is a data set containing win-loss pair of responses $\{y_w, y_l\}$ given prompt x.

B Related Work

Alignment under Preference models Most existing approaches adopt the Bradley-Terry (BT) preference model [Bradley and Terry, 1952, Christiano et al., 2017], which involves first learning a preference model and then optimizing the objective function with a KL divergence penalty relative to the original language model. For example, RLHF [Ouyang et al., 2022] aims to ensure that LLMs follow instructions by initially learning a BT model and subsequently fine-tuning the model based on the learned reward while regularizing it with the original LLM.

Building on this framework, Rafailov et al. [2024] introduces Direct Preference Optimization (DPO) that maintains the assumption of the BT model for preferences but eliminates the preference learning step by reformulating the objective and optimizing it directly. Additionally, Ethayarajh et al. [2024] diverges from the traditional BT-based methods by deriving algorithms that bypass the preference modeling step altogether. Instead, they model user preferences based on Kahneman and Tversky's utility theory.

Alignment Solution Concepts under General Preferences Azar et al. [2024] are the first to consider general preferences and propose a family of optimization objectives that optimize a function of the preferences probabilities regularized by the KL divergence with respect to the original model. They propose the IPO algorithm, an offline algorithm that directly optimizes the win rate of the model penalized by the KL divergence with respect to the original model. Munos et al. [2024] also consider general preferences and aim to find the *von Neumann winner*, which corresponds to the Nash equilibrium of a game played between the two LLMs over the win rate. They propose a variant of the Mirror Descent (MD) algorithm called Nash-MD and show last-iterate convergence in the KL-regularized game. Concurrently, Swamy et al. [2024] study the same solution concept focusing more on sequential games. Calandriello et al. [2024] proved that the objective of the the IPO algorithm coincides with the Nash policy under a proper choice of the parameter that controls the regularization.

Iterative Self-Play Algorithms Apart from the aforementioned works, a line of recent work also propose practical implementation of the Mirror Dscent (MD) algorithms, which can be used to learn the Nash equilibrium via self-play.Rosset et al. [2024] propose Direct Nash Optimization (DNO), where at each iteration, the model regresses the predicted preferences against the actual preferences using cross-entropy loss. Similarly, Wu et al. [2024] introduce the Self-Play Preference Optimization (SPPO) method, Gao et al. [2024] introduce Reinforcement Learning via Regressing Relative Rewards (REBEL), and Richemond et al. [2024] introduce the Direct Reward Optimization (DRO) which regresses the loss using the L_2 distance at each iteration. Since these algorithms simulate the MD update, when applied in a (unregularized) zero-sum game, they only have average-iterate convergence but all *diverge in last iterate*. Moreover, all these methods require the estimation of the win rate, which can be computationally intensive and may introduce estimation errors.

Most closely related to our work is Iterative Nash Policy Optimization (INPO) by Zhang et al. [2024], which continues to use L_2 distance regression. However, by further reformulating and simplifying the objective similar to IPO, INPO eliminates the need to estimate the expected win rate. The primary distinction between our approach and INPO is that INPO is designed for the KL-regularized game and is equivalent to MD; while our algorithm LINE-PO is inspired by the Conceptual Prox algorithm and guarantees last-iterate convergence in the unregularized game. This fundamental difference allows LINE-PO to achieve more favourable convergence properties with robust alignment (i.e., 50% against any other policy) for large language models.

Last-Iterate Convergence on Games It is well-established that Mirror Descent fails to converge in simple zero-sum games, often resulting in cycling behavior [Mertikopoulos et al., 2018]. In contrast, several prominent algorithms have been shown to achieve last-iterate convergence including the Proximal Point (PP) method [Rockafellar, 1976], Extra-Gradient (EG) [Korpelevich, 1976], Optimistic Gradient Descent (OGD) [Popov, 1980, Rakhlin and Sridharan, 2013], and the Conceptual Prox/Mirror Prox methods [Nemirovski, 2004]. The asymptotic convergence properties of these algorithms have been extensively studied [Popov, 1980, Facchinei and Pang, 2003, Iusem et al., 2003, Nemirovski, 2004, Daskalakis and Panageas, 2018]. Recently, there has been a growing focus on establishing finite-time convergence guarantees for these methods, addressing the practical necessity of understanding their performance within a limited number of iterations (see e.g. [Mokhtari et al., 2020b,a, Golowich et al., 2020b,a, Bauschke et al., 2021, Wei et al., 2021, Cai et al., 2022, Gorbunov et al., 2022] and references therein).

C Properties of the Prox Operator

Recall that $\mathrm{Prox}(z, g) = \mathrm{argmax}_{z' \in \mathcal{Z}} \langle g, z' \rangle - D_\varphi(z'||z) = \mathrm{argmax}_{z' \in \mathcal{Z}} \langle g + \nabla\varphi(z), z' \rangle - \varphi(z')$. The following properties of the prox operator are well-known in the literature(e.g., [Nemirovski, 2004])

Lemma 1. $\mathrm{Prox}(z, g) = z'$ *if and only if* $\langle g + \nabla\varphi(z) - \nabla\varphi(z'), z' - z^* \rangle \geq 0$ *for all* $z^* \in \mathcal{Z}$.

Corollary 1. *Let* $\mathrm{Prox}(z, g) = z'$*, then*

$$\langle g, z^* - z' \rangle \leq D_\varphi(z^*||z) - D_\varphi(z^*||z') - D_\varphi(z'||z), \quad \forall z^* \in \mathcal{Z}$$

D Proof of Theorem 1

The proof of Theorem 1 is relatively standard in the literature [Facchinei and Pang, 2003, Nemirovski, 2004]. We include a formal proof here for completeness. In Theorem 1, we make the following assumption.

Assumption 1. *We assume there exists a Nash equilibrium* $\pi^\star$ *such that* $\mathrm{supp}(\pi^\star) = \mathrm{supp}(\pi_{\mathrm{sft}})$.

This assumption is mild and **much weaker** than the “Bounded Log Density" assumptions used in previous works [Rosset et al., 2024, Zhang et al., 2024], which requires $|\log \frac{\pi^t}{\pi_{\mathrm{sft}}}|$ is bounded.

Recall that $\Pi := \{\pi : \mathrm{supp}(\pi) \subseteq \mathrm{supp}(\pi_{\mathrm{sft}})\}$. Then $\mathrm{KL}(\pi||\pi_{\mathrm{sft}}) \leq D := \max_{y:\pi_{\mathrm{sft}}(y)>0} \log \pi_{\mathrm{sft}}(y)$ is bounded for any $\pi \in \Pi$. We first prove $\mathrm{KL}(\pi^\star||\pi^{t+1}) \leq \mathrm{KL}(\pi^\star||\pi^t)$ for any $t \geq 1$.

In the proof, we assume that each step of LINE-PO, $\pi^{t+1} \leftarrow \mathrm{argmax}_{\pi_1} \min_{\pi_2} J_\tau(\pi_1, \pi_2, \pi_{\mathrm{ref}})$ can be solved exactly. Our proof extends to the case the optimization problem is solved approximately with sufficient accuracy.

Lemma 2. *Let* $\pi^\star$ *be an Nash equilibrium of* $J(\pi_1, \pi_2)$*. Then for any* $\tau > 0$*, if*

$$(\pi, \pi) = \underset{\pi_1}{\mathrm{argmax}}\, \underset{\pi_2}{\mathrm{argmin}}\, J_\tau(\pi_1, \pi_2, \pi_{\mathrm{ref}}),$$

then

$$\mathrm{KL}(\pi^\star||\pi) \leq \mathrm{KL}(\pi^\star||\pi_{\mathrm{ref}}) - \mathrm{KL}(\pi||\pi_{\mathrm{ref}})$$

Proof. By definition of the prox operator, we have

$$\begin{aligned}\pi &= \operatorname*{argmax}_{\pi_1} J_\tau(\pi_1, \pi, \pi_{\text{ref}}) \\ &= \operatorname*{argmax}_{\pi_1} \mathbb{P}(\pi_1 \succ \pi) - \tau\, \mathrm{KL}(\pi_1, \pi_{\text{ref}}) \\ &= \mathrm{Prox}(\pi_{\text{ref}}, \frac{1}{\tau}\mathbb{P}(\cdot \succ \pi)). \end{aligned} \tag{6}$$

Using Corollary 1, we have for any $\pi' \in \Pi$,

$$\frac{1}{\tau}(\mathbb{P}(\pi' \succ \pi) - \mathbb{P}(\pi \succ \pi)) \leq \mathrm{KL}(\pi' || \pi_{\text{ref}}) - \mathrm{KL}(\pi' || \pi) - \mathrm{KL}(\pi || \pi_{\text{ref}}). \tag{7}$$

Plugging $\pi' = \pi^\star$ into the above inequality and noting that $\mathbb{P}(\pi \succ \pi) = \frac{1}{2}$, we get

$$\frac{1}{\tau}\left(\mathbb{P}(\pi^\star \succ \pi) - \frac{1}{2}\right) \leq \mathrm{KL}(\pi^\star || \pi_{\text{ref}}) - \mathrm{KL}(\pi^\star || \pi) - \mathrm{KL}(\pi || \pi_{\text{ref}}).$$

Since $\pi^\star$ is a Nash equilibrium and thus $\mathbb{P}(\pi^\star \succ \pi) \geq \frac{1}{2}$, the lefthand side of the above inequality is ≥ 0. The we have

$$\mathrm{KL}(\pi^\star || \pi) \leq \mathrm{KL}(\pi^\star || \pi_{\text{ref}}) - \mathrm{KL}(\pi || \pi_{\text{ref}}).$$

□

Lemma 2 implies the following properties on the trajectory $\{\pi^t\}$.

Corollary 2. *In LINE-PO, we have*

1. $\mathrm{KL}(\pi^\star || \pi^{t+1}) \leq \mathrm{KL}(\pi^\star || \pi^t)$ *for all* $t \geq 1$.
2. $\sum_{t=1}^{\infty} \mathrm{KL}(\pi^{t+1} || \pi^t) \leq \mathrm{KL}(\pi^\star || \pi_{\text{sft}}) < +\infty$.
3. $\mathrm{supp}(\pi^t) = \mathrm{supp}(\pi_{\text{sft}})$ *for all* $t \geq 1$.

Proof. The first item is direct from Lemma 2. The second item is also direct by applying Lemma 2 for $t = 1, 2 \ldots$:

$$\sum_{t=1}^{\infty} \mathrm{KL}(\pi^{t+1} || \pi^t) \leq \sum_{t=1}^{\infty} \mathrm{KL}(\pi^\star || \pi^t) - \mathrm{KL}(\pi^\star || \pi^{t+1}) \leq \mathrm{KL}(\pi^\star || \pi_{\text{sft}}) \leq D < \infty.$$

For the third item, let $\pi^\star$ be a Nash equilibrium such that $\mathrm{supp}(\pi^\star) = \mathrm{supp}(\pi_{\text{sft}})$ as guaranteed by Assumption 1. On one hand, since $\mathrm{KL}(\pi^t || \pi^{t-1}) < \infty$ for all $t \geq 1$, we have $\mathrm{supp}(\pi^t) \subseteq \mathrm{supp}(\pi^{t-1}) \subseteq \ldots \subseteq \mathrm{supp}(\pi_{\text{sft}})$. On the other hand, $\mathrm{KL}(\pi^\star || \pi^t) < \infty$ implies $\mathrm{supp}(\pi^\star) \subseteq \mathrm{supp}(\pi^t)$. Since $\mathrm{supp}(\pi_{\text{sft}}) = \mathrm{supp}(\pi^\star)$, we have $\mathrm{supp}(\pi^t) = \mathrm{supp}(\pi_{\text{sft}}) = \mathrm{supp}(\pi^\star)$. □

Since the sequence $\{\pi^t\}$ is bounded (all lies in the simplex), it has at least one limit point $\hat{\pi}$. The next lemma shows that a limit point must be a Nash equilibrium.

Lemma 3. *If $\hat{\pi}$ is a limit point of $\{\pi^t\}$, then $\hat{\pi}$ is a Nash equilibrium of $J(\pi_1, \pi_2)$.*

Proof. By item 2 in Corollary 2, we have $\lim_{t\to\infty} \mathrm{KL}(\pi^{t+1} || \pi^t) = 0$. This implies $\lim_{t\to\infty} \|\pi^{t+1} - \pi^t\| = 0$. As $\hat{\pi}$ is a limit point of $\{\pi^t\}$, we let $\{\pi^k : k \in \kappa\}$ be the subsequence that converges to $\hat{\pi}$. Then by Equation (6), we have

$$\begin{aligned} \lim_{k\in\kappa, k\to\infty} \pi^{k+1} &= \lim_{k\in\kappa, k\to\infty} \mathrm{Prox}(\pi^k, \frac{1}{\tau}\mathbb{P}(\cdot \succ \pi^{k+1})) \\ \Rightarrow \hat{\pi} &= \mathrm{Prox}(\hat{\pi}, \frac{1}{\tau}\mathbb{P}(\cdot \succ \hat{\pi})). \end{aligned}$$

Thus $\hat{\pi}$ is a fixed point of $\mathrm{Prox}(\pi, \frac{1}{\tau}\mathbb{P}(\cdot \succ \pi)$. Moreover, by item 3 in Corollary 2, we have $\mathrm{supp}(\hat{\pi}) = \mathrm{supp}(\pi_{\text{sft}})$. Now consider both the max and min player running MWU initialized with $\pi^1 = \hat{\pi}$. Then we have $\pi^t = \hat{\pi}$ for all $t \geq 1$. By Equation (7), we have for any $\pi' \in \Pi$,

$$\frac{1}{\tau}\sum_{t=1}^{\infty}\left(\mathbb{P}(\pi' \succ \hat{\pi}) - \frac{1}{2}\right) \leq \mathrm{KL}(\pi' || \hat{\pi}) < \infty,$$

where the inequality holds since $\text{supp}(\pi') \subseteq \text{supp}(\hat{\pi})$. As a result, we get

$$\mathbb{P}(\pi' \succ \hat{\pi}) \leq \frac{1}{2}, \forall \pi' \in \Pi \Leftrightarrow \mathbb{P}(\hat{\pi} \succ \pi') \geq \frac{1}{2}, \forall \pi' \in \Pi$$

Thus $\hat{\pi}$ is a Nash equilibrium of $J(\pi_1, \pi_2)$. □

Proof of Theorem 1 Since $\hat{\pi}$ is a Nash equilibrium, by Corollary 2, $\{\text{KL}(\hat{\pi}||\pi^t) \geq 0\}$ is a decreasing sequence. Thus $\{\text{KL}(\hat{\pi}||\pi^t)\}$ converges. As a result,

$$\lim_{t\to\infty} \text{KL}(\hat{\pi}||\pi^t) = \lim_{k\in\kappa, k\to\infty} \text{KL}(\hat{\pi}||\pi^k) = \text{KL}(\hat{\pi}||\hat{\pi}) = 0.$$

Thus we have $\lim_{t\to\infty} \pi^t = \hat{\pi}$ is a Nash equilibrium. This completed the proof of Theorem 1.

E Proof of Theorem 2

We show that MWU has linear convergence to the unique Nash equilibrium of a KL-regularized zero-sum game $J(\pi_1, \pi_2, \pi_{\text{ref}})$.

We denote $\mu^\star = \pi^\star_\tau$ the unique Nash equilibrium of the KL regularized game $J\tau(\pi_1, \pi_2, \pi_{\text{ref}})$. We note that $J(\pi_1, \pi_2)$ is 1-smooth. We then can adapt [Abe et al., 2024, Lemma F.1] to our setting.

Lemma 4 (Adapted from Lemma F.1 in Abe et al. [2024])**.** *If we choose $\eta \in (0, \frac{2\tau}{3\tau^2+8}]$, then we have for every $k \geq 1$*

$$\text{KL}(\mu^\star, \mu^{k+1}) \leq (1 - \frac{\eta\tau}{2})\,\text{KL}(\mu^\star, \mu^k).$$

Applying the lemma recursively implies $\text{KL}(\mu^\star||\mu^{k+1}) \leq (1 - \frac{\eta\tau}{2})^k\,\text{KL}(\mu^\star||\pi_{\text{ref}})$ and completes the proof.

F Computing the Prox Operator using Preference Learning Methods

Reinforcement Learning algorithms We can use the Proximal Policy Optimization (PPO) algorithm [Schulman et al., 2017] to solve $\text{Prox}(\pi, \eta g)$. Observe that

$$\begin{aligned}\text{Prox}(\pi, \eta g) &= \operatorname*{argmax}_{\pi'}\{\langle \eta g, \pi' \rangle - \text{KL}(\pi'||\pi)\} \\ &= \operatorname*{argmax}_{\pi'} \mathbb{E}_{y\sim\pi'}\left[g[y] - \eta^{-1} \cdot \text{KL}(\pi'||\pi)\right]\end{aligned}$$

shares the same form as the objective in (4). Typically, we parameterize $\pi' = \pi_\theta$ with neural network parameters θ and optimize over θ.

Loss minimization algorithms Let us denote $\hat{\pi}$ the prox operator $\text{Prox}(\pi, \eta g)$, then we have

$$\hat{\pi}[y] = \frac{\pi(y)\exp(\eta g(y))}{Z} \Leftrightarrow \log \frac{\hat{\pi}(y)}{\pi(y)} - \eta g(y) + \log Z = 0,$$

where $Z = \mathbb{E}_{y\sim\pi}[\exp(\eta g(y))]$ is the partition function. We can directly compute the partition function Z and thus $\hat{\pi}$ in small tabular cases. However, the partition function is hard to compute in general large-scale applications. Several works have recently proposed to solve the above equality by optimizing the corresponding L_2 loss. Specifically, the Self-Play Preference Optimization (SPPO) loss [Wu et al., 2024] assumes $\log Z = \frac{\eta}{2}$ and optimizes

$$\ell_{\text{SPPO}}(\theta) = \left(\log \frac{\pi_\theta(y)}{\pi(y)} - \eta g(y) - \frac{\eta}{2}\right)^2.$$

The Direct Reward Optimization (DRO) loss [Richemond et al., 2024] parameterizes both $\hat{\pi}$ and $\log Z$ with θ and V_ϕ respectively and optimize[5]

$$\ell_{\text{DRO}}(\theta, \phi) = \left(\log \frac{\pi_\theta(y)}{\pi(y)} - \eta g(y) - \eta V_\phi\right)^2.$$

[5]we modified some constants in the original DRO loss to make it consistent with our presentation. The modification has no other effects.

The REBEL loss [Gao et al., 2024] uses *differences in rewards* to eliminate the partition function Z and optimize the regression loss

$$\ell_{\text{REBEL}}(\theta) = \left(\eta^{-1}\left(\log \frac{\pi_\theta(y)}{\pi(y)} - \log \frac{\pi_\theta(y')}{\pi(y')}\right) - (g(y) - g(y'))\right)^2.$$

All the above approaches can be used to solve $\text{Prox}(\pi, \eta g)$. However, directly applying them iteratively on $J(\pi_1, \pi_2)$ is equivalent to running MWU, which provably diverges. In contrast, we can apply them in Algorithm 2 and then apply our meta-algorithm LINE-PO to guarantee convergence to a Nash equilibrium with robust alignment.

Remark 1. *The above approaches are versatile and work well for any g that can be evaluated efficiently. In particular, we should consider using them when (1) $g = r$ is a reward function and we can efficiently query r; (2) $g = \mathbb{P}(\cdot \mid \mu)$ is the win rate against a reference policy μ, and we can efficiently sample from μ and have oracle access to $\mathbb{P}$. These two setting are popular and practical in the LLM alignment setting.*

Now we turn attention to the more specific setting where g corresponds to a preference model $\mathbb{P}$ (could be a BT model or a general preference) and that we can collect a win-loss preference data set $\mathcal{D} = \{(y_w, y_l)\}$, which is standard for LLM alignment. Although the abovementioned algorithms apply, they all require estimating g (the win rate) and may be inefficient in practice. In the following, we present algorithms directly working on the sampled dataset $\mathcal{D}$ without further estimation.

Sampled loss based on the BT preference model Assume $g = r$ is the reward of the Bradley-Terry model, and the dataset $\{(y_w, y_l)\}$ consists of win-lose pairs of responses. Then we can solve $\text{Prox}(\pi, \eta g)$ by optimize the DPO loss [Rafailov et al., 2024] defined as

$$\ell_{\text{DPO}}((y_w, y_l); \theta) = -\log \sigma\left(\eta^{-1} \log \frac{\pi_\theta(y_w)}{\pi(y_w)} - \eta^{-1} \log \frac{\pi_\theta(y_l)}{\pi(y_l)}\right).$$

Sampled loss for general preference The DPO loss inspires many other loss functions that work under even weaker assumptions on the preference model. Now, we assume a general preference model $\mathbb{P}$ over $\mathcal{Y}$ (not necessarily the BT model). We assume g is the win-rate against some policy μ such that $g_\mu(y) = \mathbb{P}[y \succ \mu] := \mathbb{E}_{y' \sim \mu}[\mathbb{P}[y \succ y']]$ (think of μ as the reference policy π_{ref} or other online policy π_t). We assume the dataset contains win-lose pairs sampled from μ: $\{y_w, y_l \sim \mu\}$. We denote the preference distribution $\lambda_{\mathbb{P}}(y, y')$ as a binary distribution:

$$\lambda_{\mathbb{P}}(y, y') = \begin{cases} (y, y') \text{ with probability } \mathbb{P}[y \succ y'] \\ (y', y) \text{ with probability } 1 - \mathbb{P}[y \succ y'] \end{cases}$$

The (population) IPO loss [Tang et al., 2024, Calandriello et al., 2024] is defined as

$$\ell_{\text{IPO}}(\theta, \mu) := \mathbb{E}_{(y_w, y_l) \sim \mu, (y^+, y^-) \sim \lambda_{\mathbb{P}}(y_w, y_l)}\left[\left(\log \frac{\pi_\theta(y^+)}{\pi(y^+)} - \log \frac{\pi_\theta(y^-)}{\pi(y^-)} - \frac{\eta}{2}\right)^2\right].$$

It has been proved that the minimizer of the $\ell_{\text{IPO}}(\theta, \mu)$ satisfies

$$\pi_\theta(y) \propto \pi(y) \exp\left(-\eta \mathbb{P}[y \succ \mu]\right) \Leftrightarrow \pi_\theta = \text{Prox}(\pi, \eta g_\mu).$$

Thus we can compute the prox operator $\text{Prox}(\pi, \eta g_\mu)$ where $g_\mu = \mathbb{P}(\cdot \succ \mu)$ by minimizing the IPO loss against policy μ.

A variant of the IPO loss applied to the regularized preference setting is the Iterative Nash Policy Optimization (INPO) loss [Zhang et al., 2024]. Here, we define g_μ^τ the gradient $\nabla_\pi J_\tau(\pi, \mu, \pi_{\text{ref}}) = \mathbb{P}(\cdot \succ \mu) - \tau \log \frac{\mu(\cdot)}{\pi_{\text{ref}}(\cdot)}$ of the regularized objective. The corresponding INPO loss is

$$\ell_{\text{INPO}} := \mathbb{E}_{(y_w, y_l) \sim \mu, (y^+, y^-) \sim \lambda_{\mathbb{P}}(y_w, y_l)}\left[\left(\log \frac{\pi_\theta(y^+)}{\pi_\theta(y^-)} - \eta\tau \log \frac{\pi_{\text{ref}}(y^+)}{\pi_{\text{ref}}(y^-)} - (1 - \eta\tau) \log \frac{\mu(y^+)}{\mu(y^-)} - \frac{\eta}{2}\right)^2\right].$$

Similarly, it has been shown that the INPO loss minimizer corresponds to the prox operator's solution $\text{Prox}(\pi, \eta g_\mu^\tau)$. Thus we can use the INPO in Algorithm 2 directly.

G Practical Implementation of Algorithms

We present an implementation of LINE-PO using the INPO [Zhang et al., 2024] as a subgame solver here. We remark that LINE-PO can also be implemented using PPO or many other preference learning algorithms, as we show in Section 3.3. Given the implementation of these existing methods, our meta-algorithm requires minimal change but archives last-iterate convergence to Nash equilibrium with robust alignment.

Algorithm 3: Practical Implementation of LINE-PO integrated with INPO (Algorithm 4)

Input: Initial policy π_{sft}, regularization $\{\tau_t > 0\}$, step size $\{\eta_t > 0\}$, number of iterations $T \geq 1$, number of inner optimization steps $\{K_t \geq 1\}$, preference oracle $\mathbb{P}$.

```
Initialize π^1, π_ref ← π_sft
for t = 1, 2, ..., T − 1 do
    π^{t+1} ← INPO(π_ref, τ_t, η_t, K_t, P)
    π_ref ← π^{t+1}
return π^T
```

Algorithm 4: INPO [Zhang et al., 2024]

Input: Reference policy π_{ref}, regularization $\tau > 0$, step size $\eta > 0$, number of rounds $K \geq 1$, preference oracle $\mathbb{P}$.

- Initialize $\mu^1 \leftarrow \pi_{\text{ref}}$
- **for** $k = 1, 2, \ldots, K-1$ **do**
  - Generate response pairs $\{y_1^{(i)}, y_2^{(i)}\}_{i=1}^n$ where $y_1^{(i)}, y_2^{(i)} \sim \mu^k$
  - Query preference oracle $\mathbb{P}$ to get preference data $\mathcal{D}_k = \{y_w^{(i)}, y_l^{(i)}\}_{i=1}^n$
  - Compute μ^{k+1} as

$$\mu^{t+1} = \operatorname*{argmin}_{\pi \in \Pi} \mathbb{E}_{(y_w, y_l) \sim \mathcal{D}_k} \ell_{\text{INPO}}(\pi)$$

$$\ell_{\text{INPO}}(\pi) := \mathbb{E}_{(y^+, y^-) \sim \lambda_{\mathbb{P}}(y_w, y_l)} \left[\left(\log \frac{\pi(y^+)}{\pi(y^-)} - \eta\tau \log \frac{\pi_{\text{ref}}(y^+)}{\pi_{\text{ref}}(y^-)} - (1 - \eta\tau) \log \frac{\mu^t(y^+)}{\mu^t(y^-)} - \frac{\eta}{2} \right)^2 \right]$$

- **return** μ^K

H Implementation of Mirror-Prox and Optimistic Multiplicative Weights Update

We note that there are other algorithms that has provable last-iterate convergence to Nash equilibrium in (unregularized) zero-sum games, including the Mirror-Prox algorithm [Nemirovski, 2004] and Optimistic Multiplicative Weights Update (OMWU) algorithm [Rakhlin and Sridharan, 2013, Syrgkanis et al., 2015, Hsieh et al., 2021]. We present practical implementations of these two algorithms in the context of LLM alignment for solving $J(\pi_1, \pi_2)$ (1), where we use preference optimization algorithms to solve the prox operator as shown in Section 3.3.

We denote the gradient $g(\pi) := \mathbb{P}(\cdot \succ \pi)$.

Mirror-Prox The Mirror-Prox algorithm [Nemirovski, 2004] initialized $\pi^1 = \pi_{\text{sft}}$ and updates in each iteration $t \geq 1$:

$$\pi^{t+\frac{1}{2}} = \text{Prox}(\pi^t, \eta g(\pi^t))$$
$$\pi^{t+1} = \text{Prox}(\pi^t, \eta g(\pi^{t+\frac{1}{2}}))$$

As we have shown in Section 3.3, we can implement Mirror-Prox using PPO/DPO/IPO/SPPO/DRO/REBEL to compute the prox operator. Specifically, we could sample from π^t and construct a preference dataset D_t and optimize certain regression loss (IPO/DRO/REBEL) to compute $\pi^{t+\frac{1}{2}} = \text{Prox}(\pi^t, \eta g(\pi^t))$. The procedure applies to the second step in each iteration. Thus we require two sampling and two optimization procedure in each iteration.

Optimistic Multiplicative Weights Update (OMWU) The OMWU algorithm [Rakhlin and Sridharan, 2013] is an optimistic variant of the MWU algorithm. Although MWU diverges in zero-sum games, it has been shown that OMWU has last-iterate convergence to Nash equilibrium [Wei et al., 2021, Hsieh et al., 2021]. Initialized with $\pi^1 = \pi^{\frac{1}{2}} = \pi_{\text{sft}}$, OMWU updates in each iteration $t \geq 1$:

$$\pi^{t+\frac{1}{2}} = \text{Prox}(\pi^t, \eta g(\pi^{t-\frac{1}{2}}))$$
$$\pi^{t+1} = \text{Prox}(\pi^t, \eta g(\pi^{t+\frac{1}{2}}))$$

Similarly, we can implement OMWU to solve $J(\pi_1, \pi_2)$ using preference methods to compute the prox operator as shown in Section 3.3. Moreover, OMWU has an equivalent update rule: initialize $\pi^1 = \pi^0 = \pi_{\text{sft}}$

$$\pi^{t+1} = \text{Prox}(\pi^t, 2\eta g(\pi^t) - \eta g(\pi^{t-1})),$$

which requires computing only one prox operator in each iteration.

We leave testing the practical performance of Mirror-Prox and OMWU for large-scale applications including LLM alignment as future works.

I Detailed Discussion on Synthetic Experiments

The sample-only setting is also more aligned with the practice. We use sufficient samples in each iteration for every algorithm. As a result, the LINE-PO performs the same as in the noiseless gradient setting, while the iterative IPO algorithm becomes equivalent to the MD algorithm. We present the results in Figure 2 and noting that summarize the results below.

- Iterative DPO: We observe that iterative DPO diverges and cycles between extreme policies (e.g., outputting y_a with probability close to 1). This is aligned with [Azar et al., 2024], where they found DPO will converge to the deterministic policy regardless of the regularization parameter in extreme preference settings. The cycling behavior of iterative DPO may be explained as follows: in each iteration, DPO converges to a nearly deterministic policy output y; then the new preference data shows that $y' \neq y$ is more preferred; finally, iterative DPO cycles over $\mathcal{Y}$ since the preference itself exhibits a cycle and there is no clear winner.
- Iterative IPO [Azar et al., 2024, Calandriello et al., 2024]: The IPO loss is a variant of the DPO loss, but it does not rely on the BT model assumption and works for a general preference model. However, as we have discussed before, (exactly) minimizing the IPO loss is equivalent to performing one mirror descent step, and thus, iterative IPO is equivalent to mirror descent up to sampling error. As a result, we observe that iterative IPO also exhibits cycling behavior.
- SPPO [Wu et al., 2024]: The SPPO algorithm is not exactly the same as MWU since SPPO assumes the partition function is always $Z = \log \frac{\eta}{2}$ which may not be the case. We observe that SPPO exhibits very similar cycling behavior as MD. We conclude that SPPO approximates MD very well in this instance and exhibits similar behavior.
- INPO [Zhang et al., 2024]: The INPO algorithm is designed for finding the Nash equilibrium of the KL regularized game $J_\tau(\pi_1, \pi_2, \pi_{\text{ref}})$. As we proved in Theorem 2, INPO does not diverge but exhibits last-iterate convergence. However, it converges to a regularized Nash equilibrium without the robust alignment property.