Beyond IID Constraints: A Novel Approach to Identity Preference Optimization

Stanford CS224N Custom Project

Amirhossein Afsharrad
Department of Electrical Engineering
Stanford University
afsharrad@stanford.edu

Abstract

Learning from human preferences is crucial in developing modern large language models. Reinforcement Learning with Human Feedback (RLHF) is a well-established method that has shown notable success in preference alignment. Direct Preference Optimization (DPO) was later introduced, offering comparable performance to RLHF with greater efficiency and ease of use. Subsequently, Identity Preference Optimization (IPO) was proposed to address some shortcomings of DPO while maintaining its benefits. However, IPO requires an IID-ness condition for prompt pairs in alignment datasets, which is often unmet in real-world applications, limiting its practicality. We propose an alternative variant, Extended IPO (xIPO), to handle scenarios where the IID-ness condition is not met. We also analyze the impact of non-IID data on IPO's performance, providing examples where IPO fails significantly with such data.

1 Key Information to include

- TA mentor: Archit Sharma
- External collaborators: Mohammad Ghavamzadeh[1], Ahmadreza Moradipari[2]
- No external mentor or project sharing

2 Introduction

Learning from human preferences is crucial for aligning large language models with human expectations. Reinforcement Learning with Human Feedback (RLHF) has been a key approach, leveraging reward models trained on human preferences to guide policy learning. However, RLHF relies on two important approximations: replacing pairwise preferences with pointwise rewards and assuming that the reward model generalizes well to new data. Direct Preference Optimization (DPO) simplifies this by learning policies directly from preference data, bypassing the reward modeling stage. Yet, DPO still depends on substituting pairwise preferences with pointwise rewards.

In this work, we propose Extended Identity Preference Optimization (xIPO), an advancement that addresses these limitations by handling non-IID data, which is often encountered in real-world applications. xIPO extends IPO, which initially provided a more general loss function for preference optimization but required the IID condition for prompt pairs. Our theoretical analysis and experiments show that xIPO maintains robust performance even when the IID assumption is violated, making it a versatile solution for preference alignment.

[1] ghavamza@amazon.com

[2] ahmadreza.moradipari@toyota.com

3 Related Work

Learning from human preferences has been extensively studied, particularly in the context of aligning large language models. Reinforcement Learning with Human Feedback (RLHF) Ouyang et al. (2022) has been a prominent approach, using methods like PPO Schulman et al. (2017) to optimize policies based on human-labeled reward functions. However, RLHF relies on approximations that can limit its effectiveness in certain scenarios.

Direct Preference Optimization (DPO) emerged to address some of these limitations by directly learning policies from preference data, eliminating the need for reward modeling Rafailov et al. (2023). DPO has shown competitive performance with RLHF on various language tasks, offering a simpler and resource-efficient alternative.

The Identity Preference Optimization (IPO) algorithm Gheshlaghi Azar et al. (2024) was introduced to further improve on DPO by addressing the pairwise preference assumption directly. IPO provides a more general loss function and has demonstrated empirical superiority over DPO in specific cases. However, IPO's reliance on IID data limits its applicability in real-world settings (Wang et al., 2023; Chen et al., 2022).

Our work builds on these foundations, proposing Extended IPO (xIPO) to handle non-IID data effectively. By extending IPO's capabilities, xIPO addresses a significant limitation in existing methods, ensuring robust performance across diverse datasets. This advancement is crucial for real-world applications where data often does not meet IID conditions, making xIPO a versatile and practical solution for preference optimization.

4 Approach

4.1 Background

A common method for policy optimization in human preference alignment is through minimizing an objective of the form

$$L(\pi) = \mathop{\mathbb{E}}_{\substack{x\sim\rho \\ y\sim\pi(\cdot|x)}} [r(x,y)] - \tau D_{\mathrm{KL}}(\pi \parallel \pi_{\mathrm{ref}}), \tag{1}$$

where x and y are the input prompt and output generation, π is the generation policy parameterized by the language model network, r is a reward function, π_{ref} is a reference policy, and τ is a hyper-parameter. In Reinforcement Learning with Human Feedback (RLHF) Ouyang et al. (2022), this objective is minimized using reinforcement learning methods such as PPO Schulman et al. (2017), with the reward function r learned from human-labeled preference data. Direct Preference Optimization (DPO) Rafailov et al. (2023) optimizes this objective using a method that does not involve directly learning a reward function, utilizing the same preference data to learn a minimizer of (1). Identity Preference Optimization (IPO) Gheshlaghi Azar et al. (2024) was later introduced, considering a more general loss of the form

$$\mathcal{L}_{\Psi}(\pi) = \mathop{\mathbb{E}}_{\substack{x\sim\rho \\ y\sim\pi(\cdot|x) \\ y'\sim\mu(\cdot|x)}} [\Psi(p^*(y \succ y'|x))] - \tau D_{\mathrm{KL}}(\pi \parallel \pi_{\mathrm{ref}}), \tag{2}$$

where Ψ can be any function, p^* is a probability function modeling human preference, $y \succ y'$ denotes the event of a generation y being preferred to y' by the human labeler, and μ is a behavior policy. Gheshlaghi Azar et al. (2024) suggests using the identity function for Ψ, leading to the IPO objective:

$$\mathcal{L}(\pi) = \mathbb{E}_{\substack{x\sim\rho \\ y\sim\pi(\cdot|x) \\ y'\sim\mu(\cdot|x)}} [p^*(y \succ y'|x)] - \tau D_{\mathrm{KL}}(\pi \parallel \pi_{\mathrm{ref}}). \tag{3}$$

For brevity, we drop the conditioning on x. Using a method similar to DPO, the IPO loss is introduced as

$$L^{\mathsf{IPO}}(\pi) = \mathop{\mathbb{E}}_{y,y'} \left[\left(h_\pi(y,y') - \tau^{-1}(p^*(y \succ \mu) - p^*(y' \succ \mu)) \right)^2 \right], \tag{4}$$

where $p^*(y \succ \mu)$ is the expected probability of a generation y being preferred to a sample generation from the behavior distribution μ,

$$p^*(y \succ \mu) = \mathop{\mathbb{E}}_{z\sim\mu} [p^*(y \succ \mu)], \tag{5}$$

and the function h_π is defined as

$$h_\pi(y, y') = \log\left(\pi(y)/\pi_{\text{ref}}(y)\right) - \log\left(\pi(y')/\pi_{\text{ref}}(y')\right). \tag{6}$$

Since p^* is not accessible, Gheshlaghi Azar et al. (2024) introduces the Population IPO loss

$$L^{\mathsf{pIPO}}(\pi) = \mathop{\mathbb{E}}_{y,y'}\left[\left(h_\pi(y, y') - \tau^{-1} I(y, y')\right)^2\right], \tag{7}$$

and the Sampled IPO loss

$$L^{\mathsf{sIPO}}(\pi) = \mathop{\mathbb{E}}_{y,y'}\left[\left(h_\pi(y_w, y_\ell) - \tau^{-1}/2\right)^2\right], \tag{8}$$

where $I(y, y')$ is an indicator variable that is 1 if y is preferred to y' and 0 otherwise. y_w and y_ℓ are the winner and loser in the comparison of y, y' by the human labeler. We drop the conditioning on x for brevity. Gheshlaghi Azar et al. (2024) proves that the IPO loss in (3) is equivalent to the losses in (7) and (8) given the constraint that y and y' are IID. This condition is often unmet in real-world applications. Thus, we propose a novel loss to handle non-IID input samples.

4.2 Identity Preference Optimization with Non-IID Samples

As stated earlier, the IPO algorithm in Gheshlaghi Azar et al. (2024) works with IID data, but cannot handle non-IID data. Our goal remains to minimize the true objective (3). However, without access to IID pairs (y, y'), minimizing the loss in (4), (7), or (8) is not feasible. We modify the IPO loss (4) for non-IID pairs y, y'. We call the new loss $L^{\mathsf{IPO}}_{\mu_1,\mu_2}$, emphasizing that the sampled generations y, y' can follow different distributions:

$$L^{\mathsf{IPO}}_{\mu_1,\mu_2}(\pi) = \mathop{\mathbb{E}}_{\substack{y\sim\mu_1 \\ y'\sim\mu_2}}\left[\left(h_\pi\left(y, y'\right) - \tau^{-1}(p^*(y \succ \mu) - p^*(y' \succ \mu))\right)^2\right]. \tag{9}$$

Though (9) suggests (y, y') are independent, they need not be, and our results hold without this assumption. We must show that optimizing (9) and (3) are equivalent, as stated in the following theorem.

Theorem 1. *For any given distributions μ_1, μ_2, μ with the same support, the loss* (9) *has a unique minimizer π^*, which is also the unique optimal policy of* (4)*.*

The proof is in Appendix A. This shows that the IID condition in (4) is unnecessary. However, this does not transfer directly to (7) or (8). Without the IID assumption, $L^{\mathsf{IPO}}(\pi)$, $L^{\mathsf{pIPO}}(\pi)$, and $L^{\mathsf{sIPO}}(\pi)$ become different functions, none equivalent. $L^{\mathsf{IPO}}(\pi)$ explicitly depends on μ, while $L^{\mathsf{pIPO}}(\pi)$ and $L^{\mathsf{sIPO}}(\pi)$ do not, if $\mu_1, \mu_2 \neq \mu$. The next theorem explains their relationship.

Theorem 2. *Let $L^{\mathsf{pIPO}}_{\mu_1,\mu_2}(\pi)$ and $L^{\mathsf{sIPO}}_{\mu_1,\mu_2}(\pi)$ be defined as*

$$L^{\mathsf{pIPO}}_{\mu_1,\mu_2}(\pi) = \mathop{\mathbb{E}}_{\substack{y\sim\mu_1 \\ y'\sim\mu_2}}\left[\left(h_\pi(y, y') - \tau^{-1} I(y, y')\right)^2\right], \quad L^{\mathsf{sIPO}}_{\mu_1,\mu_2}(\pi) = \mathop{\mathbb{E}}_{\substack{y\sim\mu_1 \\ y'\sim\mu_2}}\left[\left(h_\pi(y_w, y_\ell) - \tau^{-1}/2\right)^2\right]. \tag{10}$$

Then,

$$L^{\mathsf{sIPO}}_{\mu_1,\mu_2}(\pi) = L^{\mathsf{IPO}}_{\mu_1,\mu_2}(\pi) + \tau^{-1} \mathop{\mathbb{E}}_{\substack{y\sim\mu_1 \\ y'\sim\mu_2}}\left[h_\pi(y, y')\right] + c, \tag{11}$$

where c is a constant independent of π.

Corollary 2.1. $\mathbb{E}_{\substack{y\sim\mu_1 \\ y'\sim\mu_2}}\left[h_\pi(y, y')\right]$ *reduces to the difference in expected log-likelihood ratios over μ_1 and μ_2. If $\mu_1 = \mu_2$, this term is zero, making $L^{\mathsf{pIPO}}(\pi)$ and $L^{\mathsf{sIPO}}(\pi)$ equivalent.*

Proof of Theorem 2 is in B. Corollary 2.1 confirms the equivalence of the losses in Gheshlaghi Azar et al. (2024), and Theorem 2 shows how this equivalence fails without the IID assumption. See Section 5.4 for simulations and examples.

We have shown that the standard IPO pipeline fails without IID data. Next, we introduce methods to address this issue.

4.3 xIPO: The Extended Identity Preference Optimization

To address the earlier challenge, we propose a novel loss function and prove its equivalence to our main objective. Our Extended IPO (xIPO) loss is defined as

$$L^{\mathsf{xIPO}}_{\mu_1,\mu_2}(\pi) = \mathop{\mathbb{E}}_{\substack{y\sim\mu_1\\ y'\sim\mu_2\\ z\sim\mu}} \left[\left(h_\pi(y,y') - \tau^{-1}\left(I(y,z) - I(y',z) \right) \right)^2 \right], \tag{12}$$

where $I(y,z)$ is 1 if the human labeler prefers y over z and 0 otherwise. Theorem 3 provides the required theoretical guarantee, showing that minimizing the new loss is equivalent to optimizing our original objective.

Theorem 3. *The extended IPO loss $L^{\mathsf{xIPO}}_{\mu_1,\mu_2}(\pi)$ introduced in* (12) *and the non-IID IPO objective $L^{\mathsf{IPO}}_{\mu_1,\mu_2}(\pi)$ defined in* (9) *are equivalent up to a constant independent of π.*

Corollary 3.1. *Minimizing $L^{\mathsf{xIPO}}_{\mu_1,\mu_2}(\pi)$ is equivalent to minimizing $\mathcal{L}(\pi)$ in* (3)*. This combines the results of Theorems 1 and 3.*

The xIPO loss handles non-IID data samples but requires triplets (y, y', z) and two comparisons, $I(y,z)$ and $I(y',z)$. The original IPO only needed pairs (y,y') and a comparison $I(y,y')$. This trade-off allows for a broader dataset but requires more data and labeling. The loss $L^{\mathsf{IPO}}_{\mu_1,\mu_2}(\pi)$ in (9) depends on three distributions μ_1, μ_2, μ, necessitating data samples from all three.

We define the sampled version of the xIPO loss (12) as

$$\widehat{L}^{\mathsf{xIPO}}(\pi) = \frac{1}{|\mathcal{D}|} \sum_{(y,y',z)\in\mathcal{D}} \left[\left(h_\pi(y,y') - \tau^{-1}\left(I(y,z) - I(y',z) \right) \right)^2 \right]. \tag{13}$$

With the sampled xIPO loss, one can use any finite dataset and network structure with parameters θ to parameterize the policy π_θ and solve the alignment task by minimizing the loss. See Section 5.4 for supporting simulations and results.

Before concluding this section, we note that the xIPO algorithm requires triplets (y, y', z) instead of pairs (y, y'). Obtaining such data can be challenging, especially since data generation is expensive. Many applications prefer using existing datasets, which generally lack the z samples needed for our approach. In the next part, we introduce a method to address this issue for cases where the original data only includes pairs (y, y') but also provides human-provided scores for each y and y'. This is inspired by the UltraFeedback dataset Cui et al. (2023), a recent and commonly used dataset for alignment tasks.

4.4 Score-based xIPO: A Simplified Version of xIPO

We note an important observation about the xIPO loss in (12) regarding generation z. The value of z is unimportant; only the comparison results $I(y,z)$ and $I(y',z)$ matter. This motivates constructing $I(y,z)$ and $I(y',z)$ without explicitly sampling $z \sim \mu$.

Inspired by the UltraFeedback dataset Cui et al. (2023), where generations y and y' have quality scores from 1 to 10, we introduce score-based xIPO. Given quality scores for y and y', this method doesn't need explicit samples of z. Suppose the score function s is given, and $s(y) \in \mathbb{R}$ is the quality score for generation y. In a simplistic model, we can assume

$$I(y,z) = \mathbb{1}(s(y) > s(z)) = \begin{cases} 1 & s(y) > s(z) \\ 0 & \text{otherwise} \end{cases}. \tag{14}$$

For simplicity, we use this deterministic model. We can rewrite $I(y,z) - I(y',z)$ as

$$I(y,z) - I(y',z) = \mathbb{1}(s(y) > s(z)) - \mathbb{1}(s(y') > s(z)). \tag{15}$$

Given a distribution μ over all possible generations z, consider the corresponding score distribution μ_s. Instead of sampling z from μ and scoring it, we directly sample a score $s_z \sim \mu_s$ and use it in the xIPO loss. This allows us to rewrite the xIPO loss (12) as

$$L^{\mathsf{xIPO}}_{\mu_1,\mu_2}(\pi) = \mathop{\mathbb{E}}_{\substack{y\sim\mu_1\\ y'\sim\mu_2\\ s_z\sim\mu_s}} \left[\left(h_\pi(y,y') - \tau^{-1}\left(\mathbb{1}(s(y) > s_z) - \mathbb{1}(s(y') > s_z) \right) \right)^2 \right]. \tag{16}$$

If μ_s is hard to calculate, one can estimate μ_s using samples $z \sim \mu$ or assume a reasonable distribution μ_s to generate scores. This method is effective because our goal is to find a policy that maximizes the average probability of outperforming policy μ while staying close to a reference policy π_{ref}. By introducing a score distribution μ_s, our policy will learn to outperform the policy μ implicitly defined by μ_s. Thus, a proper choice of μ_s can lead to a well-performing final policy.

5 Experiments

We ran the following two sets of experiments:

1. **Bandit experiments.** To test our theoretical claims and new fine-tuning methods, we first experimented with bandit problems with relatively small action sets. In a bandit problem, there is no input prompt (x), and the policy chooses an action from a finite set of actions.
2. **Preference alignment with LLMs experiments.** We tested our fine-tuning algorithms using the UltraFeedback Cui et al. (2023) dataset and compared the results with baselines. For policy parameterization, we used the Tule-2 Ivison et al. (2023) model.

Each subsection below details these experimental settings.

5.1 Data

5.1.1 Bandit experiments

For the bandit experiments, we used synthetic data constructed locally. For a bandit problem with N actions ($[N] = \{1, \cdots, N\}$), the distributions μ_1, μ_2, μ are discrete probability vectors of length N. For the xIPO experiments, triplets (y, y', z) are sampled according to these probability vectors. Comparisons $I(y, z)$ and $I(y', z)$ use a Bradley-Terry model, with a reward vector $r \in \mathbb{R}^N$ and $I(y, z)$ sampled from a Bernoulli distribution given by $\mathbb{P}\left[I(y, z) = 1\right] = \mathbb{P}\left[y \succ z\right] = \sigma\left(r(y) - r(z)\right)$, where $\sigma(q) = 1/(1 + \exp(-q))$ is the sigmoid function. The Bradley-Terry model is not required for our algorithms but is used here for its popularity in preference modeling.

For the score-based experiments, scores are defined from 0 to 10 (following UltraFeedback Cui et al. (2023)). For an action $i \in [N]$, the corresponding score is $s(i) = 10i/N \in [\frac{10}{N}, 10]$.

5.1.2 Preference alignment with LLMs experiments

We used the UltraFeedback dataset Cui et al. (2023), which includes about 64,000 instruction prompts, each with four responses generated by different models from a pool of 17 models including GPT-4, GPT-3.5, Bard, LLaMA2-7B/13B/70B-Chat, Alpaca-7B, and Falcon-40B-Instruct. Each response is scored from 1 to 10 by GPT-4. This dataset provides our (y, y') entries. To generate the z values, we use the Tulu-2 7B SFT model[3]. UltraFeedback instructions are inputs for generating z values, which are then scored using the same GPT-4 scoring pipeline used by UltraFeedback.

5.2 Evaluation Method

5.2.1 Bandit experiments

For the bandit experiments, we have synthetic data, giving us access to all environment parameters. This allows us to compute the optimal policies analytically. We use the distance from the optimal policy as our evaluation metric, comparing policies from pIPO, sIPO, and xIPO.

5.2.2 Preference alignment with LLMs experiments

For response generation, we use the standard win-rate metric. We calculate the win-rate of each model's output compared to the four responses from the UltraFeedback dataset. The responses are scored by GPT-4. Using the same scoring pipeline, we score our generations. By comparing scores, we determine the win-rate against the best and worst of the four UltraFeedback generations for pIPO, sIPO, xIPO, and score-based xIPO.

[3] `https://huggingface.co/allenai/tulu-2-dpo-70b`

5.3 Experimental Details

5.3.1 Bandit experiments

The bandit experiment parameters are given in Table 1. Since there is no input prompt, the network has N parameters, and the output is generated by passing these parameters through a sigmoid layer to produce action probabilities or a policy.

# actions	# samples	τ	# epochs	Learning rate
3	10,000	0.1	20,000	2×10^{-3}

Table 1: Experiment parameters

5.3.2 Preference alignment with LLMs experiments

Training parameters and structures are the same as those available on the AllenAI GitHub repository[4]. The only modification is the change in loss functions. We kept all other elements of the fine-tuning pipeline the same and fine-tuned the Tulu-2 SFT 7B model using pIPO, sIPO, and xIPO objectives.

5.4 Results

5.4.1 Bandit experiments

Loss functions. We consider a bandit setting with an action set of size 3. This means there are only three possible outputs and one context, and each policy π can be represented as a vector of length 3, with non-negative entries that sum to 1. Thus, we can parameterize each policy π by two parameters x and y. In other words, we can write $\pi = [x \quad y \quad 1 - x - y]$. This allows us to express any loss function $L(\pi)$ such as those in (3), (4), (7), and (8) as 3-dimensional plots. Here, the loss $L(\pi) = L(x, y)$ is represented as a function of the two independent variables x and y over the region $\{(x, y) | x \geq 0, y \geq 0, x + y \leq 1\}$.

Figure 1 shows the plots of $L^{\mathsf{IPO}}(\pi)$, $L^{\mathsf{xIPO}}_{\mu_1,\mu_2}(\pi)$, and $L^{\mathsf{sIPO}}_{\mu_1,\mu_2}(\pi)$. The first two coincide (up to an additive constant) and are depicted in Figure 1a, while the latter is presented in Figure 1b. The parameters used for these simulations are:

$$p^*(0 \succ 1) = 0.8, \quad p^*(1 \succ 2) = 0.8, \quad p^*(0 \succ 2) = 0.2, \quad \tau = 0.1,$$
$$\mu_1 = (0.90, 0.05, 0.05), \quad \mu_2 = (0.05, 0.90, 0.05), \quad \mu = (0.05, 0.05, 0.90),$$

with π_{ref} chosen to be uniform.

Figure 1 shows how different the minimizers of these loss functions are. This means that when the condition $\mu_1 = \mu_2 = \mu$ is not met, using the vanilla IPO from Gheshlaghi Azar et al. (2024) can result in a policy far from the optimal. However, since $L^{\mathsf{IPO}}(\pi)$ and $L^{\mathsf{xIPO}}_{\mu_1,\mu_2}(\pi)$ coincide, our xIPO loss is equivalent to the true objective, allowing convergence to the optimal policy.

Training results. Figure 2 shows the distance of the optimal policy from the policies achieved by each algorithm over time. For all variants of IPO *i.e.*, pIPO, sIPO, and xIPO, the optimal policy is the same and given by

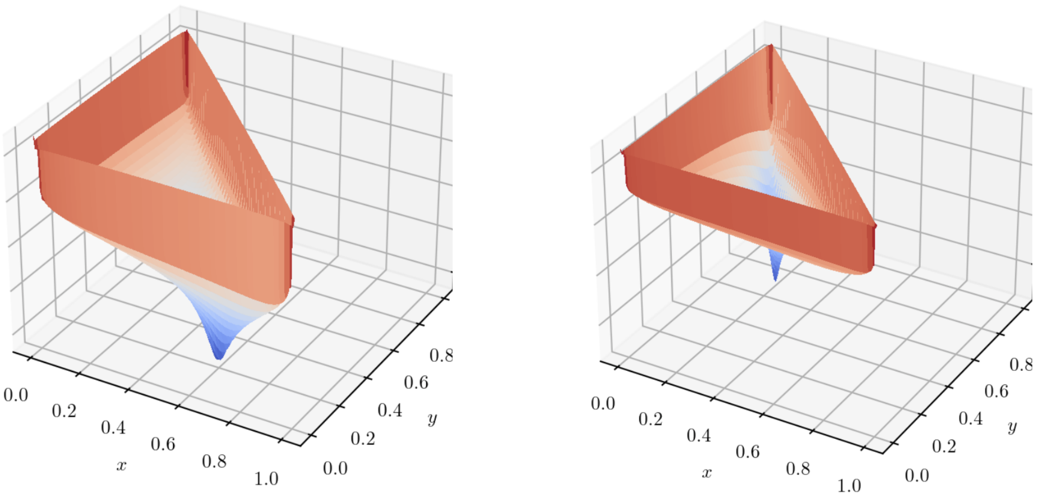
$$\pi^*(y) \propto \pi_{\text{ref}}(y) \exp\left(\tau^{-1} p^*(y \succ \mu)\right),$$

while the DPO optimal policy requires a notion of reward and the Bradley-Terry model assumption and is different. To generate the plot in Figure 2, we used these analytical optimal policies and computed their distances from the policies generated by the algorithms sIPO, xIPO, pIPO, and DPO.

As the plot in Figure 2 shows, only DPO and our extension of IPO, xIPO, converge to the optimal policy, as their distance from the optimal policy converges to zero. The other two variants of IPO do not achieve the optimal solution. This confirms our earlier results and supports our claim that when the distributions of the generation pair (y, y') are not the same, vanilla IPO does not necessarily work. We need to use our new method, extended IPO, which works perfectly.

Score-based xIPO. Score-based xIPO, as detailed in Section 4.4, is a variant of xIPO that, given access to quality scores for all output generations, implements xIPO without the need to explicitly generate samples $z \sim \mu$. The theoretical implications and guarantees of xIPO remain unchanged, as score-based xIPO has the same loss function but minimizes it differently. The results from the previous sections hold for score-based xIPO as well. Appendix D includes sample results for brevity.

[4] `https://github.com/allenai/open-instruct`



(a) True IPO loss and extended IPO loss

(b) Sampled IPO loss

Figure 1: Comparison of true IPO loss, sampled IPO loss, and extended IPO losses

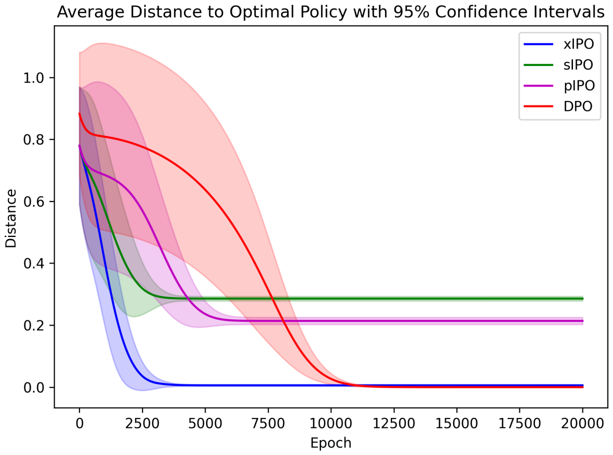


Figure 2: Distance to optimal policy for different preference alignment algorithms

5.4.2 Preference alignment with LLMs experiments

We compare the outcome of pIPO, sIPO, and xIPO with the responses in the UltraFeedback dataset that have the highest and lowest responses, using GPT-4 as the proxy human labeler to determine the winner and loser, and hence the win-rate. The results are presented in Table 2.

	pIPO	**sIPO**	**xIPO**
Win-rate against best UltraFeedback response (%)	48.4	57.8	58.4
Win-rate against worst UltraFeedback response (%)	73.2	82.1	83.7

Table 2: Experiment results

pIPO shows the worst performance, which is expected as pIPO is the initial version of IPO presented in Gheshlaghi Azar et al. (2024), and sIPO was proposed as a more stable version. Comparing xIPO and sIPO, we observe a slight improvement with xIPO, but not as much as expected. This may be because the UltraFeedback dataset is close to being IID, making it less suitable for comparing xIPO and sIPO. Additionally, our evaluation metric is win-rate based on human preference, which is not always aligned with the optimal policy of (4). While xIPO performs similarly to sIPO, it might still be closer to π^* of (4), aligning with our theoretical results.

6 Analysis

We observed that xIPO generalizes the performance of IPO for non-IID input data. Comparing pIPO and xIPO losses reveals that both aim to fit $h_\pi(y, y')$ to a constant independent of π. In pIPO, this constant can be 0 or 1, corresponding to y being better or worse than y'. However, for xIPO, there are three cases: -1, 0, or 1, as the difference $I(y, z) - I(y', z)$ can take these values. This gives xIPO more flexibility in comparisons, allowing for win, lose, and draw, determined by z or the behavior policy μ. This provides intuition on how xIPO makes IPO more general.

xIPO has higher variance than sIPO because sIPO uses asymmetry in pIPO to reduce variance. This trick is not applicable to xIPO as it is symmetric, so no further symmetrization is possible. Therefore, if the IID assumption holds, sIPO might perform better than xIPO due to its lower estimator variance.

An important consideration is the choice of μ. The goal, as defined by (4), is to find a policy that performs better than μ. If μ is an extremely good policy, running xIPO to outperform it might result in $I(y, z) = I(y', z) = 0$ over the dataset, since z is always better than y and y'. The model might learn that z is better than y and y' without necessarily learning how to be better than z, unless it sees examples that are better than z. This is crucial when selecting μ based on the available training data.

7 Conclusion

In this project, we explored the limitations of Identity Preference Optimization (IPO) with non-IID data and proposed Extended IPO (xIPO) as a solution. Our theoretical analysis and experiments demonstrated that xIPO generalizes IPO to handle non-IID data effectively, maintaining the ability to converge to the optimal policy. We showed that xIPO, unlike pIPO and sIPO, accommodates comparisons that include ties, providing a more flexible framework for preference alignment. Our experiments with synthetic bandit problems confirmed that xIPO and DPO converge to the optimal policy, while pIPO and sIPO do not, validating our theoretical claims. In the context of large language models, we observed that xIPO offers a slight improvement over sIPO, although the dataset used may not fully highlight the benefits of xIPO due to its near-IID nature.

Despite these achievements, our work has limitations. xIPO exhibits higher variance than sIPO, which may affect its performance when the IID assumption holds. Additionally, the choice of the behavior policy μ is critical, as selecting an extremely good policy could lead to suboptimal learning outcomes. Future work could focus on further reducing the variance of xIPO and exploring more sophisticated methods for selecting μ. Additionally, applying xIPO to a broader range of datasets and real-world applications would help validate its effectiveness and generalize its use cases. Investigating alternative models to the deterministic score-based comparisons and integrating more realistic scenarios could also enhance the robustness of xIPO.

8 Ethics Statement

This project is not about a specific application of language models. Instead, we are providing a new general fine-tuning method so that people can use a larger group of datasets for their fine-tuning purposes. This means that everyone can use our proposed fine-tuning method for their own specific application. As a result, our project is unfortunately challenged by almost every ethical challenge that may exist. For instance, one important problem with our project, which improves how language models learn from data, is that these models might pick up and even increase biases from the data they are trained on. This means they could unfairly favor or discriminate against certain groups of people. Another risk that comes to mind is that people might use these improved models to create misleading or harmful content because they can write very well. To handle these issues, we can check the data and the model's outputs to find and lessen these biases. We will look at how the model's responses vary across different groups of people and make adjustments where necessary. However, we believe that as hard as we try, one can still overpower our considerations if they have bad intentions. So, one of the best ways could be to produce a set of rules on how to use our model responsibly, making it clear what it should and shouldn't be used for, and act based on trust.

Acknowledgements

The author has kindly been advised by Mohammad Ghavamzadeh (Amazon Research), Faisal Ladhak (Amazon Research), and Ahmadreza Moradipari (Toyota Research). We have used compute resources of Amazon and Toyota for parts of the simulations in this work. However, except for the final execution of the LLM simulations on Toyota servers, all other parts of the project have solely been done by the author.

References

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

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 4447–4455. PMLR.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms.

A Proof of Theorem 1

The proof is almost exactly the same as the proof of Theorem 2 in the IPO paper Gheshlaghi Azar et al. (2024), with the only difference being the change in the positive multiplicative weights in Equation (15) of Gheshlaghi Azar et al. (2024). The rest of the proof is the same.

B Proof of Theorem 2

Consider the difference $L^{\mathsf{sIPO}}_{\mu_1,\mu_2}(\pi) - L^{\mathsf{IPO}}_{\mu_1,\mu_2}(\pi)$. Expanding the quadratic terms in both expectations, the terms $h_\pi(y,y')^2$ cancel out. Also, the constant terms are not important and will construct the constant c. Thus, we need to show that the difference of the corss-terms amounts to $\mathbb{E}_{\substack{y\sim\mu_1\\ y'\sim\mu_2}}[h_\pi(y,y')]$, *i.e.*,

$$\frac{2}{\tau}\mathbb{E}_{\substack{y\sim\mu_1\\ y'\sim\mu_2}}[h_\pi(y,y')I(y,y')] - \frac{1}{\tau}\mathbb{E}_{\substack{y\sim\mu_1\\ y'\sim\mu_2}}[h_\pi(y_w,y_\ell)] = \frac{1}{\tau}\mathbb{E}_{\substack{y\sim\mu_1\\ y'\sim\mu_2}}[h_\pi(y,y')],$$

or equivalently,

$$\mathbb{E}_{\substack{y\sim\mu_1\\ y'\sim\mu_2}}[h_\pi(y_w,y_\ell)] = 2\mathbb{E}_{\substack{y\sim\mu_1\\ y'\sim\mu_2}}[h_\pi(y,y')I(y,y')] - \mathbb{E}_{\substack{y\sim\mu_1\\ y'\sim\mu_2}}[h_\pi(y,y')]. \tag{17}$$

$$\begin{aligned}
\mathop{\mathbb{E}}_{\substack{y\sim\mu_1\\y'\sim\mu_2}}\left[h_\pi(y_w,y_\ell)\right] &= \mathop{\mathbb{E}}_{\substack{y\sim\mu_1\\y'\sim\mu_2}}\left[h_\pi(y,y')p^*(y\succ y')+h_\pi(y',y)p^*(y'\succ y)\right]\\
&= \mathop{\mathbb{E}}_{\substack{y\sim\mu_1\\y'\sim\mu_2}}\left[h_\pi(y,y')p^*(y\succ y')-h_\pi(y,y')\left(1-p^*(y'\succ y)\right)\right]\\
&= \mathop{\mathbb{E}}_{\substack{y\sim\mu_1\\y'\sim\mu_2}}\left[2h_\pi(y,y')p^*(y\succ y')-h_\pi(y,y')\right]\\
&= 2\mathop{\mathbb{E}}_{\substack{y\sim\mu_1\\y'\sim\mu_2}}\left[h_\pi(y,y')p^*(y\succ y')\right]-\mathop{\mathbb{E}}_{\substack{y\sim\mu_1\\y'\sim\mu_2}}\left[h_\pi(y,y')\right]\\
&= 2\mathop{\mathbb{E}}_{\substack{y\sim\mu_1\\y'\sim\mu_2}}\left[h_\pi(y,y')I(y,y')\right]-\mathop{\mathbb{E}}_{\substack{y\sim\mu_1\\y'\sim\mu_2}}\left[h_\pi(y,y')\right]
\end{aligned} \tag{18}$$

C Proof of Theorem 3

It suffices to show the equality of the cross terms in the two loss functions. This is given by

$$\begin{aligned}
\mathop{\mathbb{E}}_{\substack{y\sim\mu_0\\y'\sim\mu_1}}\left[h_\pi(y,y')\left(p^*(y\succ\mu)-p^*(y'\succ\mu)\right)\right] &= \mathop{\mathbb{E}}_{\substack{y\sim\mu_0\\y'\sim\mu_1}}\left[h_\pi(y,y')\mathop{\mathbb{E}}_{z\sim\mu}\left[p^*(y\succ z)-p^*(y'\succ z)\right]\right]\\
&= \mathop{\mathbb{E}}_{\substack{y\sim\mu_0\\y'\sim\mu_1\\z\sim\mu}}\left[h_\pi(y,y')\left(p^*(y\succ z)-p^*(y'\succ z)\right)\right]\\
&= \mathop{\mathbb{E}}_{\substack{y\sim\mu_0\\y'\sim\mu_1\\z\sim\mu}}\left[h_\pi(y,y')\left(I(y,z)-I(y',z)\right)\right].
\end{aligned} \tag{19}$$

D Score-based xIPO simulation results

We have considered bandit examples with $N=30$ actions. For the score-based experiments, scores are defined from 0 to 10. For an action $i\in[N]$, the corresponding score is $s(i)=10i/N\in[\frac{10}{N},10]$. For two different score distributions μ_s, we have plotted the score distributions, and the final policies given by DPO, (sampled) IPO, and xIPO. Figures 3, 4, 5, 6 show the first scenario, and Figures 7, 8, 9, 10 demonstrate the second.

We can observe that the score distribution has a direct effect on all the algorithms. Specifically, xIPO tends to converge to policies that outperform the policy μ that is implicitly defined by the score distributions. However, once it outperforms that policy, it will stop the optimization and does not further optimize over the absolute score values.

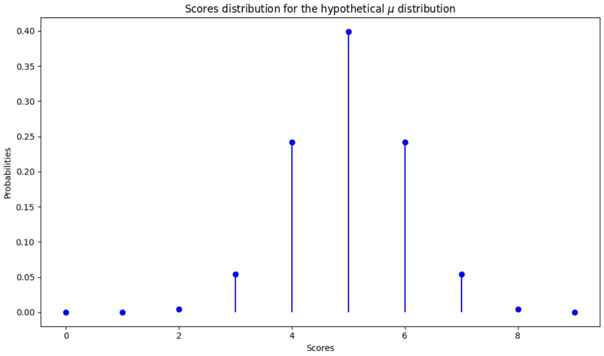


Figure 3: Score distributions – case 1

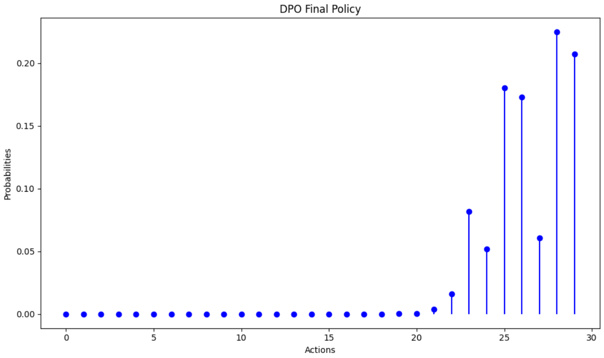


Figure 4: DPO policy – case 1

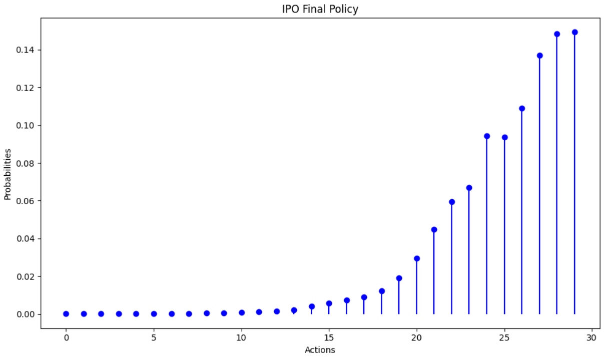


Figure 5: IPO policy – case 1

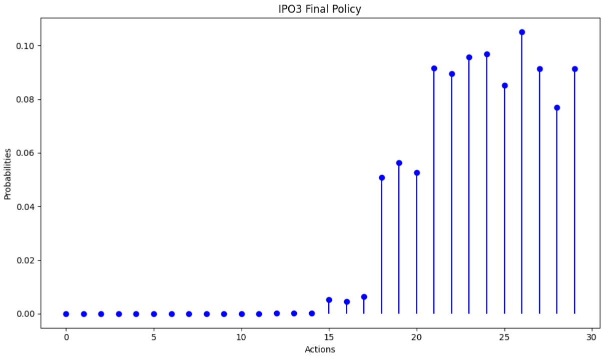


Figure 6: xIPO policy – case 1

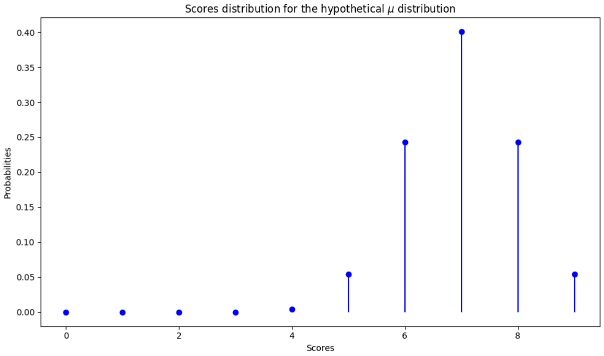


Figure 7: Score distributions – case 2

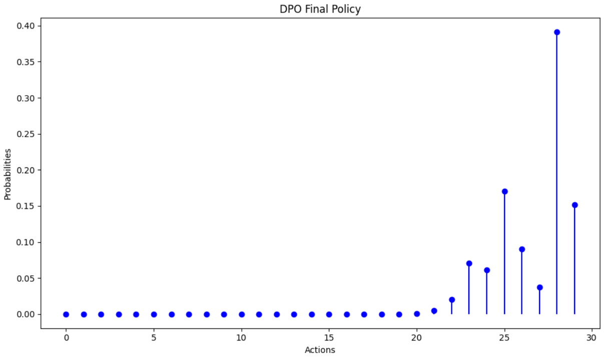


Figure 8: DPO policy – case 2

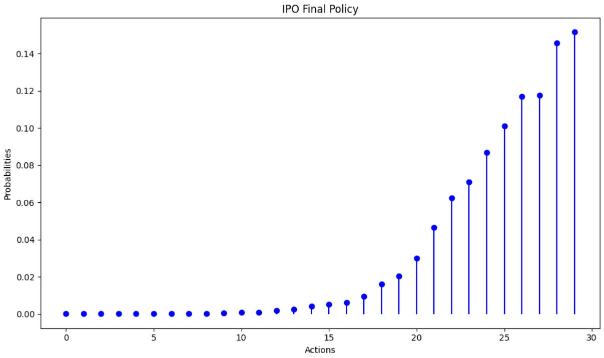


Figure 9: IPO policy – case 2

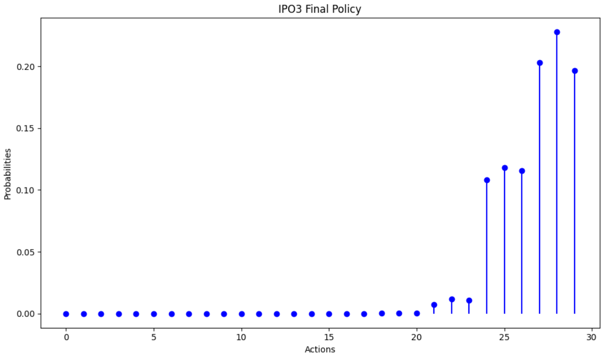


Figure 10: xIPO policy – case 2