A Minimaximalist Approach to Reinforcement Learning from Human Feedback

Gokul Swamy [1] **Christoph Dann** [2] **Rahul Kidambi** [2] **Zhiwei Steven Wu** [1] **Alekh Agarwal** [2]

Abstract

We present *Self-Play Preference Optimization* (SPO), an algorithm for reinforcement learning from human feedback. Our approach is *minimalist* in that it does not require training a reward model nor unstable adversarial training and is therefore rather simple to implement. Our approach is *maximalist* in that it provably handles non-Markovian, intransitive, and stochastic preferences while being robust to the compounding errors that plague offline approaches to sequential prediction. To achieve the preceding qualities, we build upon the concept of a *Minimax Winner* (MW), a notion of preference aggregation from the social choice theory literature that frames learning from preferences as a zero-sum game between two policies. By leveraging the symmetry of this game, we prove that rather than using the traditional technique of dueling two policies to compute the MW, we can simply have a *single* agent play against itself while maintaining strong convergence guarantees. Practically, this corresponds to sampling multiple trajectories from a policy, asking a *preference* or teacher model to compare them, and then using the proportion of wins as the reward for a particular trajectory. We demonstrate that on a suite of continuous control tasks, we are able to learn significantly more efficiently than reward-model based approaches while maintaining robustness to the intransitive and stochastic preferences that frequently occur in practice when aggregating human judgments.

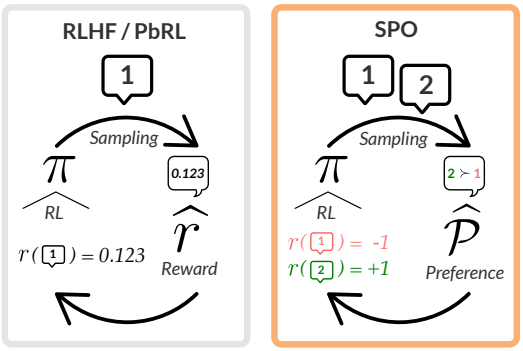


Figure 1: The standard pipeline (left) for preference-based RL / RLHF involves training a *reward* model (i.e. a classifier) based on a dataset of pairwise preferences and then optimizing it via RL. We introduce SPO (right), a method that instead optimizes directly based on preference feedback provided by a *preference* or teacher model, with each trajectory getting a reward based on the proportion of other on-policy trajectories it is preferred to. We prove and validate empirically that this approach is more robust to intransitive, non-Markovian, and noisy preferences than prior works.

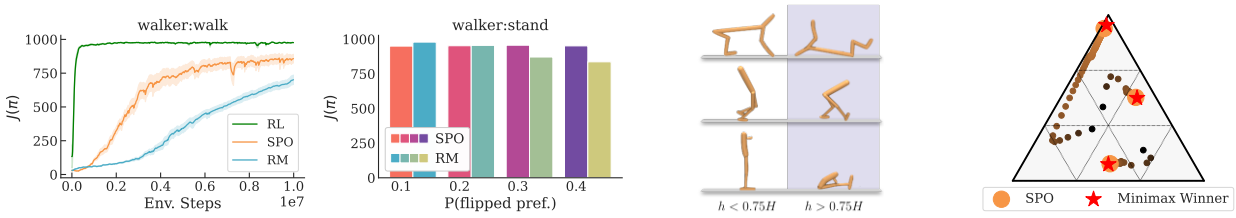
1. Introduction

Reinforcement learning from human feedback (RLHF, Christiano et al. (2017)) also known as preference-based reinforcement learning (PbRL, Akrour et al. (2012); Wirth et al. (2017); Sadigh et al. (2017); Ibarz et al. (2018); Lee et al. (2021b;a); Sikchi et al. (2022)), is a technique for policy optimization based on relative, rather than absolute, feedback. Owing to the relative ease of providing comparative feedback rather than absolute scores for agent behavior for human raters (Miller, 1956), RLHF has been successfully applied across fields from robotics (Zucker et al., 2011; Cakmak et al., 2011; Tucker et al., 2020; Swamy et al., 2020; Bıyık et al., 2020) to recommendation (De Gemmis et al., 2009; Ailon & Mohri, 2010; Viappiani & Boutilier, 2010; Afsar et al., 2022), to retrieval (Yue & Joachims, 2009). As of late, RLHF has attracted renewed interest as a leading technique for fine-tuning large language models (LLMs) (Ziegler et al., 2020; Stiennon et al., 2020; Bai et al., 2022a; Ouyang et al., 2022).

The predominantly studied approach to RLHF is via *Reward-*

Most of this work was completed while GS was a Student Researcher at Google Research. [1]Carnegie Mellon University [2]Google Research. Correspondence to: Gokul Swamy <gswamy@cmu.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.



(a) Transitive Preferences (b) Stochastic Preferences (c) Non-Markovian Preferences (d) Intransitive Preferences

Figure 2: SPO gives us a unified approach to optimize a variety of preference structures. Starting with the "nicest" case where preferences are consistent with a single reward signal ***(a)***, SPO is more sample-efficient than iterative Reward Modeling (RM) based approaches. Moreover, SPO learns comparable to RM-based approaches with stochastic preferences, where we flip preferences corresponding to a ground truth reward, without the burden of an extra model ***(b)***. Furthermore, SPO handles complex non-Markovian preferences ***(c)*** such as learning to maximize rewards over the first three-quarters of a trajectory while not crossing a threshold on returns in the last quarter, despite searching over a class of Markovian policies. Lastly, considering intransitive preferences generated by aggregating sub-populations where a single reward function cannot explain the preferences, SPO computes a Minimax Winner (MW) consistently across problem instances ***(d)***. In Fig. 5(a), we show that the *RM based approach fails* in this setting and always converges to a deterministic policy.

based RLHF, a two-stage procedure. First, given pairs of preferred and dis-preferred behavior, one trains a *reward model* to assign higher scores to the former via a classification objective. One then optimizes this reward function via some reinforcement learning algorithm.

Simple as the above recipe is, the key ingredient of a reward model can have some undesirable effects. First, assuming an underlying reward function exists is equivalent to assuming that there exists a *total order* over agent behavior. This means that there are no intransitivities in rater preferences (i.e. $A \succ B, B \succ C \Rightarrow A \succ C$), which contradicts what psychology tells us about actual human decision making (Tversky, 1969; Gardner, 1970). Even if one believes an *individual* person's preferences are transitive, when aggregated across a *population* of raters, as is necessary at scale, transitivity is unlikely to be satisfied (May, 1954). Second, given the inherent stochasticity of human preferences (Agranov & Ortoleva, 2017), one often learns a reward model that leads to a collapse in generation diversity. For example, consider a problem where the agent can pick one of two options, each of which is preferred by a sub-population of raters that makes up half of the total population. Due to either finite sample or optimization error, we can easily learn a model that assigns a slightly higher reward to one option over the other. Then, if we were to optimize our policy under this model, we would learn to (almost) exclusively select one option, leaving half of the population unsatisfied.

In recognition of the above concerns, various reward-model-free approaches have been proposed in the prior literature. One particularly promising set of techniques frames RLHF as a two-player zero-sum game between two policies, each of which attempts to produce behavior that is preferred by a rater to the other's (Yue et al., 2012). While elegant theoretically, this "dueling" framing inherits the inherent instability of adversarial training in practice and has therefore mostly been applied to bandit problems (Dudík et al., 2015; Saha et al., 2021; Saha & Krishnamurthy, 2022) (with some recent exceptions: Wang et al. (2023); Munos et al. (2023)).

Motivated by these issues, we provide a simple, theoretically rigorous, and empirically performant approach to RLHF that eliminates reward modeling and does not require adversarial training. Our approach follows from two key insights. First, ***by framing RLHF as a two-player zero-sum game, we are able to truly eliminate reward models*** and therefore are robust to noisy, intransitive, and non-Markovian preferences that frequently occur in practice. Second, ***by leveraging the symmetry of the game, we prove that we can simply train a single agent in a self-play fashion***, eliminating the need for unstable adversarial training. Practically, this corresponds to sampling multiple trajectories from the agent, asking a *preference* or teacher model to compare each pair, and setting the reward to be the trajectory's win rate. We call our approach **SPO**: Self-Play Preference Optimization. More explicitly, our contributions are as follows:

1. We derive SPO: an algorithm for RLHF that avoids reward modeling, compounding errors, and adversarial training. By building upon the concept of a *Minimax Winner* from social choice theory, we are able to frame RLHF as a two-player zero-sum game. We then leverage the symmetry of the payoff matrix of this game to prove that we can simply train a single agent against itself.

2. We use a reduction-based analysis to investigate the convergence properties of SPO. When intransitive preferences exist, we prove that SPO converges to an approximate Minimax Winner at the rate of the underlying no-regret algorithm. We also prove that when an underlying reward function does exist, our approach converges to the optimal policy at a fast rate that matches that of standard techniques.

3. We demonstrate that on a suite of continuous control tasks with realistic preference functions, SPO is more performant than reward-model based approaches. We find that our approach is able to learn more sample-efficiently than reward-model based approaches across a

Approach	Example	Compounding Errors	Intransitive Prefs	Learning Setup
Offline, Reward-Based	DPO (Rafailov et al., 2023)	✗, Example D.1	✗, Theorem 2.4	Offline, log-loss
Online, Reward-Based	PPO (Ouyang et al., 2022)	✓	✗, Theorem 2.4	Online RL
Online, Dueling	DBGD (Yue & Joachims, 2009)	✓	✓	Online *adversarial* RL
Online, Preference-Based	**SPO** (ours)	✓	✓	Online RL

Table 1: An taxonomy of RLHF algorithms and the sorts of issues they are robust to.

variety of preference setups. This includes trajectory-level comparisons based on ground-truth Markovian rewards in the easiest case, stochastic preferences, trajectory-level non-Markovian preferences and intransitive preferences induced by aggregating over sub-populations. The strong performance of SPO in the latter three challenging setups, all motivated by practical situations, is illustrated in Figure 2.

Due to limited space, we discuss related work in detail in Appendix A. Table 1 describes the relationships between and relative benefits of the different approaches to RLHF.

2. Reinforcement Learning from Human Feedback via Game Solving

We begin by introducing the notation we will use throughout the paper before defining our solution concept and deriving an efficient algorithm to compute it.

2.1. Preliminaries

Consider a finite-horizon reward-free Markov Decision Process (MDP) (Puterman, 2014) parameterized by $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, H \rangle$ where $\mathcal{S}$, $\mathcal{A}$ are the state and action spaces, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition operator, and H is the horizon. [1] We use $\Xi \triangleq (\mathcal{S} \times \mathcal{A})^H$ to denote the space of trajectories and $\Phi_h \triangleq \times(\mathcal{S} \times \mathcal{A})^{h-1} \times \mathcal{S}$ to denote the space of histories of length h.

Preference Oracle. In the preference-based RL setup, we are given query access to a *preference function*

$$\mathcal{P} : \Xi \times \Xi \to [-1, 1] \tag{1}$$

which, given two trajectories $\xi_1, \xi_2 \in \Xi \times \Xi$, outputs a scalar that indicates the preferred trajectory. Explicitly, given some comparison function $P(\xi_1 \succ \xi_2)$, we define $\mathcal{P}(\xi_1, \xi_2) = 2P(\xi_1 \succ \xi_2) - 1$. Practically, this could be either be a preference model trained on an offline dataset (RLAIF, Bai et al. (2022b); Munos et al. (2023); Zhao et al. (2023)) or a human-in-the-loop (RLHF, Tucker et al. (2020)). The former setup is similar to reward-based RLHF, except that the reward model learning step is replaced by learning a pairwise preference model, which is more natural when learning from pairwise preference data. [2] Viewed this way, the preference-model based RLHF based methodology strictly generalizes reward-based RLHF, as we can still represent a preference function that is induced by a difference of rewards, but not every preference function is expressible this way, as we illustrate in the following sections. Also, in the "alignment" of generative models the preference function sometimes consists of a prompted generative model that is asked to compare two outputs, rather than a model trained explicitly on a preference dataset (Bai et al., 2022b). In this setting, we are able to optimize directly based on the outputs of such a model, rather than requiring a reward model detour – **see Figure 4 for a full workflow.**

By construction, preference functions are anti-symmetric, i.e. $\forall \xi_1, \xi_2 \in \Xi \times \Xi$, $\mathcal{P}(\xi_1, \xi_2) = -\mathcal{P}(\xi_2, \xi_1)$. Similarly, we have that $\forall \xi \in \Xi$, $\mathcal{P}(\xi, \xi) = 0$.

We assume access to a convex and compact policy class $\Pi \subseteq \{\mathcal{S} \to \Delta(\mathcal{A})\}$. With a slight abuse of notation, we can now define the preference function over policy pairs as

$$\mathcal{P}(\pi_1, \pi_2) \triangleq \mathbb{E}_{\xi_1 \sim \pi_1, \xi_2 \sim \pi_2}[\mathcal{P}(\xi_1, \xi_2)]. \tag{2}$$

2.2. A Brief Introduction to Social Choice Theory

Given choices from a population of raters that are represented as a preference function $\mathcal{P}$, social choice theory (Sen, 1986) studies the question of how best to select options that satisfy the diversity of preferences inherent in the said population. For example, consider the set of preferences $\mathcal{P}_1$ over options (a, b, c, d) in Figure 3.

	a	b	c	d
a	0	+1	+1	−1
b	−1	0	+1	−1
c	−1	−1	0	+1
d	+1	+1	−1	0

Figure 3: An intransitive preference function $\mathcal{P}_1$ over (a, b, c, d). $\mathcal{P}_1(x, y) = 1$ if $P(x \succ y) = 1$, -1 if $P(x \succ y) = 0$, and 0 if $P(x \succ y) = 0.5$. Observe that there is no unique Copeland Winner.

Given this preference function, perhaps the most natural idea would be to pick the option that beats the largest number

[1] We omit contexts for simplicity of presentation but they can be added without much overhead, as we detail in Appendix C.8.

[2] e.g. $\widehat{\mathcal{P}} = \operatorname{argmin}_{\widetilde{\mathcal{P}}} \mathbb{E}_{(\xi^+, \xi^-) \sim \mathcal{D}}[-\log(\frac{1}{2}(\widetilde{\mathcal{P}}(\xi^+, \xi^-) + 1)]$.

of other options. In the above matrix, this would be either option a or d as they have the largest row sums. More formally, this technique is known as a *Copeland Winner* and can be expressed mathematically as

$$\mathsf{CW}(\mathcal{P}) \triangleq \operatorname*{argmax}_{\pi \in \Pi} \sum_{\pi' \in \Pi} \mathcal{P}(\pi, \pi'). \tag{3}$$

While intuitively appealing, Copeland Winners are often not unique as in our above example, raising the question of how to break ties. For example, if half of the group feels like $a \succ d$ and the other half like $d \succ a$, picking either option would leave half of the group unsatisfied. This problem only gets worse as the number of options to choose between increases, as there is unlikely to be a single option that *everyone* prefers to *every* other option (Dudík et al., 2015).[3]

In essence, approaches that train reward models like reward-based RLHF (or implicitly assume them like DPO) are akin to computing Copeland Winners. Observe that our above matrix has an *intransitivity*: $a \succ c, c \succ d, d \succ a$. This means that *no* reward function can explain the above preferences as it would need to satisfy $r(a) > r(c)$, $r(c) > r(d)$ and $r(d) > r(a)$ simultaneously, an impossibility. Thus, the model is forced to tie-break between a and d, potentially leaving half of the population rather unsatisfied. In practice, this tie-breaking is performed based on the incredibly noisy data used to train the reward model (Taori et al., 2023; Touvron et al., 2023), making it entirely arbitrary. When combined with the fact that an ϵ difference in reward model outputs can lead to an entirely different optimal policy, we are left with an unsatisfying solution.

One potential solution to the issues with the Copeland Winner is to *randomize*. For example, we could attempt to pick a distribution over options such that we prefer samples from this distribution to those from any other distribution with probability at least $\frac{1}{2}$. For $\mathcal{P}_1$, this would correspond to us picking (a, c, d), each with probability $\frac{1}{3}$, as

$$\min_{z \in \{a,b,c,d\}} (\mathcal{P}_1(a, z) + \mathcal{P}_1(c, z) + \mathcal{P}_1(d, z))/3 = 0.$$

Intuitively, this means that while we don't always make everyone happy (an impossibility, Arrow (1950); Satterthwaite (1975)), we never pick a solution that makes a significant portion of the population consistently unhappy. Readers familiar with game theory might recognize that the above corresponds to computing the Nash equilibrium of the two-player zero-sum (2p0s) game with payoffs given by the preference function. Formally, we define the *Minimax Winner* (MW, Kreweras (1965); Simpson (1969); Kramer (1973); Fishburn (1984)), also known as a *von Neumann Winner* (Dudík et al., 2015), as the following pair of strategies:

$$\mathsf{MW}(\mathcal{P}) \triangleq \big(\operatorname*{argmax}_{p \in \Delta(\Pi)} \min_{q \in \Delta(\Pi)} \mathbb{E}_{\pi_1 \sim p, \pi_2 \sim q}[\mathcal{P}(\pi_1, \pi_2)], \operatorname*{argmin}_{q \in \Delta(\Pi)} \max_{p \in \Delta(\Pi)} \mathbb{E}_{\pi_1 \sim p, \pi_2 \sim q}[\mathcal{P}(\pi_1, \pi_2)] \big).$$

Via Sion's minimax theorem (Sion, 1958), we can guarantee that the above solution concept *always* exists, unlike a unique Copeland Winner. We note that because we assumed Π is convex, we are *always* able to collapse down any distribution $p \in \Delta(\Pi)$ to a *single* policy $\widetilde{p} \in \Pi$ while preserving solution quality by performing a weighted average:

$$\widetilde{p}(\xi) = \sum_{\pi \in \Pi} p(\pi) p(\xi | \pi). \tag{4}$$

In short, this means that we never need to explicitly maintain a distribution over policies in practice. We conclude with a few observations about MWs. First, nowhere in defining a MW did we need to assume the existence of an underlying reward function, rendering the above solution concept *truly* reward model-free. Second, in the case where there actually does exist an underlying reward function that explains the observed preferences, the MW coincides with the optimal policy for that reward (Dudík et al., 2015), rendering the MW a strict generalization of the CW. Third, MWs satisfy a variety of desirable *consistency* properties (e.g. merging populations that agree on a MW cannot change the outcome, which is especially important when attempting to re-use preference datasets), which deterministic options like the CW cannot satisfy simultaneously (Brandl et al., 2016). We now turn our attention to efficiently computing MWs.

2.3. One Player is All You Need for RLHF

Efficient algorithms for computing Nash equilibria of 2p0s games are a central focus in computational game theory. A popular approach is to run two no-regret algorithms (e.g. Hedge (Freund & Schapire, 1997) or Online Gradient Descent (Zinkevich, 2003)) against each other, known more commonly as adversarial training (Goodfellow et al., 2014). When applied to computing MWs, this is known as *dueling* (Yue et al., 2012) and is commonly applied in the bandit setting. While elegant in theory, this technique inherits all of the notorious instabilities of adversarial training when function approximation is introduced. Even ignoring optimization issues, simply storing both models in memory might be difficult in our current era of "foundation" models (Bommasani et al., 2021). These issues may explain why dueling techniques have seen limited practical use.

In light of the above difficulties, we ask a simple question: ***do we actually need two players to compute MWs?*** We prove rigorously that we only need a *single* player due to the anti-symmetry of preference functions. All proofs for this section can be found in Appendix C.

[3] For example, we see empirical evidence of this point in the high rates of inter-annotator disagreement (Taori et al., 2023; Touvron et al., 2023) in LLM finetuning datasets.

First, we prove that there *always* exists a symmetric MW.

Lemma 2.1. $\exists(\hat{p}, \hat{q}) \in \mathsf{MW}(\mathcal{P})$ *s.t.* $\hat{p} = \hat{q}$. *[Proof]*

Next, we prove that we can compute a symmetric MW by running a single no-regret algorithm against its own iterates. We assume access to the following optimization oracle.

Definition 2.2. $\mathcal{O}$ is a *no-regret online linear optimization algorithm over* $\Delta(\Pi)$ if it produces iterates $p_{t+1} = \mathcal{O}(\ell_{1:t}) \in \Delta(\Pi)$ such that, for any sequence of T linear loss functions of the form $\ell_t(p) = \mathbb{E}_{\pi\sim p}[f_t(\pi)] \in [-1, 1]$ (with $f_t : \Pi \to [-1, 1]$), we have

$$\sum_{t=1}^{T} \ell_t(p_t) - \min_{p^\star \in \Delta(\Pi)} \sum_{t=1}^{T} \ell_t(p^\star) \leq \mathsf{Reg}(T), \tag{5}$$

with $\lim_{T\to\infty} \frac{\mathsf{Reg}(T)}{T} = 0$.

Common algorithms like gradient descent satisfy this property (Zinkevich, 2003). See Hazan et al. (2016) for a more extensive list. We define the **SPO Loss** at round $t \in [T]$ as the negative preference against the current iterate p_t,

$$\ell_t^{\texttt{SPO}}(p) \triangleq \mathbb{E}_{\pi\sim p, \pi'\sim p_t}[-\mathcal{P}(\pi, \pi'))]. \tag{6}$$

We can now state our main result.

Theorem 2.3. *Consider a single copy of an algorithm* $\mathcal{O}$ *which satisfies Definition 2.2. Initialize* $p_1 \in \Delta(\Pi)$ *and set* $p_{t+1} = \mathcal{O}(\ell_{1:t}^{SPO})$. *Then,* $\bar{p} = (p_1 + \ldots + p_T)/T$ *is a* $\frac{2\mathsf{Reg}(T)}{T}$*-approximate Minimax Winner. [Proof]*

For ease of presentation, Theorem 2.3 is stated and proved assuming *full feedback*, where we observe $\ell_t^{\texttt{SPO}}(p_t)$ at round t. In Appendix C.7, we generalize the argument to the more realistic *bandit feedback* setting, where we only observe a preference $\mathcal{P}(\pi, \pi')$ for some π, π'.

Proof Sketch. Consider a pair of strategies, $p, q \in \Delta(\Pi)$. Define $\ell_t^1(p) = \mathbb{E}_{\pi\sim p, \pi'\sim q_t}[-\mathcal{P}(\pi, \pi'))]$ and $\ell_t^2(q) = \mathbb{E}_{\pi\sim p_t, \pi'\sim q}[\mathcal{P}(\pi, \pi'))]$. By the results of Freund & Schapire (1997), we know that updating $p_{t+1} = \mathcal{O}(\ell_{1:t}^1)$ and $q_{t+1} = \mathcal{O}(\ell_{1:t}^2)$ implies that average strategies $\bar{p} = \frac{1}{t}\sum_i^t p_i$, $\bar{q} = \frac{1}{t}\sum_i^t q_i$ converge to a Nash equilibrium (Minimax Winner) at the rate of the underlying no-regret algorithm. By construction, we set $p_0 = q_0$. This implies that $\ell_0^1(p) = \mathbb{E}_{\pi\sim p, \pi'\sim q_0}[-\mathcal{P}(\pi, \pi'))] = \mathbb{E}_{\pi\sim p, \pi'\sim p_0}[-\mathcal{P}(\pi, \pi'))] = \mathbb{E}_{\pi\sim p_0, \pi'\sim p}[-\mathcal{P}(\pi, \pi'))] = \ell_0^2(p)$ due to the anti-symmetry of $\mathcal{P}$. We now proceed by induction. If, at some time τ, p_τ and q_τ have the same strategy and perform updates based on the same loss function, for any deterministic $\mathcal{O}$, $p_{\tau+1} = q_{\tau+1}$. For randomized $\mathcal{O}$, our argument still applies as we describe in the full proof. Then, by the anti-symmetry of $\mathcal{P}$, we have that $\ell_{\tau+1}^1 = \ell_{\tau+1}^2$. Hence, by induction, we have that $p_t = q_t, \forall t \in [T]$. □

The above result implies that if we run our algorithm for long enough, we can get arbitrarily close to an exact MW.[4] Observe that we didn't need to assume we were running a particular algorithm $\mathcal{O}$, rendering the above a *reduction* of computing minimax winners to no-regret online learning.[5]

The above update can also be viewed from the perspective of a *dynamic reward model*: it is equivalent to performing an RL step with a ***policy-dependent reward model***:

$$r_t^{\texttt{SPO}}(\xi) \triangleq \mathbb{E}_{\pi'\sim p_t}[\mathbb{E}_{\xi'\sim\pi'}[\mathcal{P}(\xi, \xi')]]. \tag{7}$$

This reward model incentivizes the learner to play trajectories that are preferred to its current distribution. Game-solving amounts to repeatedly taking a small step along this direction before (implicitly) updating the reward model. Thus, one can view SPO as using a perfectly shaped *curriculum* to gently guide the learner. We now pause and further contextualize our results by considering a few questions.

Q1: ***Do reward-based RLHF algorithms also compute MWs?*** We prove that this is not the case in general by analyzing multiple algorithms which assume that there exists a reward function that explains the observed preferences, even if it is not maintained explicitly, e.g., as in Direct Preference Optimization (DPO, Rafailov et al. (2023)).

Theorem 2.4. *There exists a* $\mathcal{P}$ *and reference policy* π_{ref} *such that the optimal policies of reward-based RLHF and DPO are not the Minimax Winner. [Proof]*

Proof Sketch. Consider a three element Π. We can construct a $\mathcal{P}$, with a unique Minimax Winner of the form $[x, x, 1-2x]$ for $x \in (\frac{1}{3}, \frac{1}{2})$. Assume $\pi_{\text{ref}} = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$. Then, reward-based methods can only pick a reward model that makes *(a)* one option preferred to all others (resulting in some permutation of $[1, 0, 0]$), *(b)* making two options preferred to the third (resulting in some permutation of $[\frac{1}{2}, \frac{1}{2}, 0]$) or *(c)* makes all equally preferred (resulting in $[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$). None of these options can represent the Minimax Winner. □

However, by appeal to Equation 7, standard RLHF algorithms / DPO applied *iteratively* with each batch of preferences collected on-policy and a sufficiently high frequency of reward model updates may be able to compute MWs on average – we explore this idea in our experiments

Q2: ***If there exists an optimal policy / Copeland Winner for my problem, could running SPO be an inefficient way to compute it?*** We prove that this is not the case in a strong sense: when there exists a clearly optimal policy, our above algorithm converges to it at a *fast statistical rate* – $\widetilde{O}(\frac{1}{T})$

[4] For last iterate (rather than average iterate) convergence, one can simply set the no-regret algorithm to be Optimistic Mirror Descent and apply the results of Daskalakis et al. (2017).

[5] We note that in contrast to the usual asymptotic convergence guarantees one gets for self-play (Leslie & Collins, 2006), we inherit the rate of the underlying no-regret algorithm.

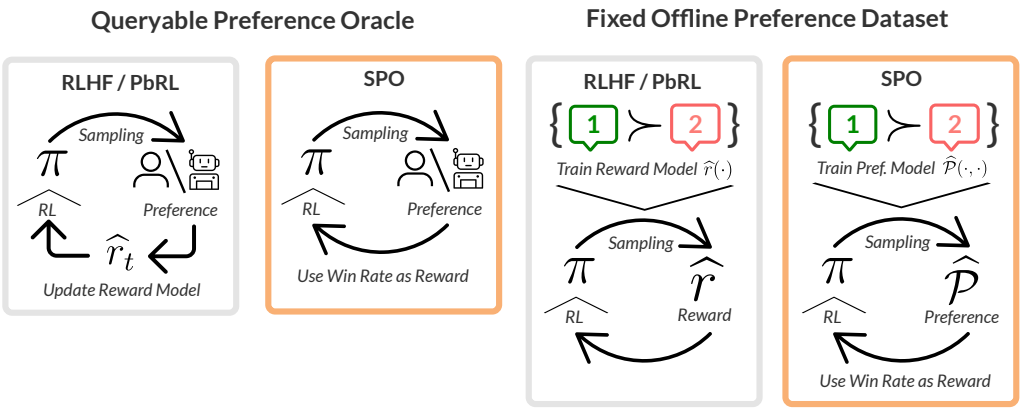


Figure 4: **Left**: In the setting where we are able to query the preference function online, SPO saves us a detour through a reward model. **Right:** In the setting where we are given a fixed dataset, we are still able to apply SPO. Rather than fit a reward model, we fit a *preference model* $\widehat{\mathcal{P}} : \Xi \times \Xi \to [-1, 1]$ that takes in both trajectories in each preference pairs and attempts to maximize the likelihood of the preferred trajectory. Intuitively, this model is modeling *relative* probabilities rather than *absolute* scores. One would then use this $\widehat{P}$ as a plug-in estimate for $\mathcal{P}$ and runs SPO as usual. We do not explore the offline dataset paradigm in our experiments.

Algorithm 1 SPO (Theoretical Version)

1: **Input:** Learning rate η, Iterations T, Preference fn. $\mathcal{P}$.
2: **Output:** Trained policy π.
3: Initialize $\pi_h(\cdot|\phi) = \mathsf{Unif}(\mathcal{A}), \forall \phi \in \Phi_h, h \in [H]$.
4: **for** t in $1 \dots T$ **do**
5: `// Preference to` π_t `as reward`
6: Compute $r_t(\xi) = \mathbb{E}_{\xi' \sim \pi_t} \mathcal{P}(\xi, \xi')$.
7: Set $Q_t^h(\phi, a) = \mathbb{E}_{\xi \sim \pi_t}[r_t(\xi)|\phi_h = \phi, a_h = a]$.
8: Set $A_t^h(\phi, a) = Q_t^h(\phi, a) - \mathbb{E}_{a' \sim \pi_t^h(\phi)}[Q_t^h(\phi, a')]$.
9: **for** $\phi, a, h \in \Phi_h \times \mathcal{A} \times [H]$ **do**
10: `// use no-regret algo for update`
11: $\pi_{t+1}^h(a|\phi) \propto \pi_t^h(a|\phi) \exp\left(\eta A_t^h(\phi, a)\right)$.
12: **end for**
13: **end for**
14: **Return** $\bar{\pi}$, trajectory-level mixture of $\pi_{1:T}$.

instead of the usual $\widetilde{O}(\frac{1}{\sqrt{T}})$ average regret. This matches the rates for UCB-style methods, previously been studied in more restricted versions of the dueling setup (Bengs et al., 2021). We give a simpler version of the result here, with a more general form and proof in Appendix C.6.

Corollary 2.5 (Informal)**.** *Suppose that there exists $\pi^* \in \Pi$ such that $\mathcal{P}(\pi^*, \pi) > \Delta$ for all $\pi \neq \pi^*$ and $-\Delta \leq \mathcal{P}(\pi, \pi') \leq \Delta$ for all $\pi, \pi' \neq \pi^*$. Then, after T rounds, the average solution $\bar{p}$ computed using Hedge as oracle $\mathcal{O}$ and ℓ_t^{SPO} is an $\widetilde{O}(\frac{|\Pi|}{\Delta T})$-approximate Minimax Winner.*

2.4. SPO: Self-Play Preference Optimization.

For single-step problems with a small and discrete policy class, it is common to maintain a distribution over policies / arms. However, as we transition to the sequential setting with a large and often continuous policy class, it is difficult to scale such an approach. We are therefore faced with the question of *what is the right no-regret algorithm to optimize the sequence of SPO losses (Equation 7) in the RL setting?*

To answer this question, we turn to the celebrated idea of *local regret minimizers* (Zinkevich et al., 2007; Even-Dar et al., 2009). Consider a problem with a finite state space. Then, at each state $s \in \mathcal{S}$, we could independently instantiate a no-regret algorithm that optimizes over $\Delta(\mathcal{A})$, feeding it a loss that depends on the cumulative reward received after exiting the state, i.e. $Q(s, a)$. Then, regardless of the state distribution our resulting policy induces, we can guarantee that we're improving at each iteration. For a specific no-regret algorithm (Hedge, Freund & Schapire (1997)), this leads to the well-known *soft policy iteration* (SPI) procedure (Ziebart, 2010). Because our reward function is at the trajectory level, we technically need to have a regret minimizer at each *history* rather than at each state. We describe in Algorithm 1 an instantiation of SPO that uses history-dependent SPI as its policy optimizer (Lines 6-10) and now present a performance guarantee on the learned policy.

Corollary 2.6. *With an appropriate setting of η, running Algorithm 1 for T iterations guarantees that $\bar{\pi}$ is a $8H\sqrt{\frac{\log(|\mathcal{A}|)}{T}}$-approximate Minimax Winner. [Proof]*

This follows directly from our Theorem 2.3.

SPO in the Contextual Bandit Setting. For certain RLHF applications (i.e. LLM preference fine-tuning), it is common to model the problem as a contextual bandit rather than a bona fide RL problem. In such settings, SPI simplifies back down to the Hedge algorithm (Freund & Schapire, 1997). This means that for all contexts (prompts) $x \in \mathcal{X}$ and all

Algorithm 2 SPO (Practical Version)

Input: Iterations T, Preference fn. $\mathcal{P}$, Queue size B, Reinforcement learning algo. $\mathsf{RL} : \Pi \times \mathcal{D} \to \Pi$.
Output: Trained policy π.
Initialize $\pi_1 \in \Pi$, Queue $\mathcal{Q} \leftarrow [\xi_{1:B} \sim \pi_1]$.
for t in $1 \dots T$ **do**
  Sample $\xi_t \sim \pi_t$.
  `// Win-rate over queue as reward`
  Compute $r_t(\xi_t) = \frac{1}{|\mathcal{Q}|} \sum_{q=1}^{B} \mathcal{P}(\xi_t, \xi_q)$.
  Set $r_t^h = r_t(\xi_t)/H, \forall h \in [H]$.
  `// use PPO, TRPO, SAC ...`
  $\pi_{t+1} \leftarrow \mathsf{RL}(\pi_t, \mathcal{D} = \{(s_t^h, a_t^h, r_t^h)\}_{h \in [H]})$.
  $\mathcal{Q} \leftarrow [\xi_2, \dots, \xi_B, \xi_t]$.
end for
Return best of $\pi_{1:T}$ on validation data.

arms (completions) $y \in \mathcal{Y}$, SPO (Algorithm 1) simplifies to

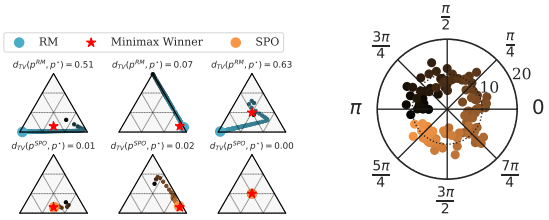
$$\pi_{t+1}(y|x) \propto \pi_t(y|x) \cdot \exp\left(\eta \mathbb{E}_{y' \sim \pi_t(x)}[\mathcal{P}(y \succ y'|x)]\right).$$

We do not explore the contextual setting in our experiments and leave a thorough study of approximations of the above update that scale to modern-day generative modeling applications to future work. See Appendix C.8 for the natural extension of Algorithm 1 the contextual MDP setting.

SPO in Continuous Control. One often uses a deep network to represent their policy rather than the tabular representation we assume in Algorithm 1. It turns out that for certain policy parameterizations, the Natural Policy Gradient (NPG) algorithm of Kakade (2001) is *exactly* equivalent to the soft policy iteration procedure (Agarwal et al., 2021). Many standard deep RL algorithms like TRPO (Schulman et al., 2015) and PPO (Schulman et al., 2017) are explicitly motivated as approximating NPG, while techniques like SAC (Haarnoja et al., 2018) can also be viewed as approximate soft policy iteration. Thus, as we move towards practice, we are free to choose from a wide set of deep RL techniques as reasonable approximations of Algorithm 1.

One other concern we need to resolve is how to do credit assignment with trajectory-level feedback. In fact, if we are unable to provide rewards for each timestep in the problem, our options for policy optimization are quite limited outside of notoriously high variance REINFORCE-style policy gradients (Williams, 1992). We suggest a simple fix to this problem: just split the trajectory-level reward equally amongst all state-action pairs. We prove in Appendix C that doing so preserves policy optimality.

Lemma 2.7. *Consider a trajectory-level reward function r and define $\Pi^\star = \operatorname{argmax}_{\pi \in \Pi} \mathbb{E}_{\xi \sim \pi}[r(\xi)]$ as the corresponding set of optimal policies. For all (s_t, a_t) in any ξ, let $\tilde{r}(s_t, a_t) = r(\xi)/H$. Then, we have that $\Pi^\star = \operatorname{argmax}_{\pi \in \Pi} \mathbb{E}_{\xi \sim \pi}[\sum_h^H \tilde{r}(s_t, a_t)]$. [Proof]*



(a) Intransitive Prefs. (Discrete) (b) Intransitive Prefs. (Ant-v3)

Figure 5: **(a):** Across a variety of intransitive preferences over a discrete set of three options, SPO learns the MW almost exactly, while RM always converges to a corner – see Appendix F.1 for details. **(b):** We visualize the position of our agent in a Mujoco Ant-v3 environment at the end of an episode over the course of training. Each dot represents 100k steps of training. We see that our agent traces out a circle of radius 10 (the MW) on average. More seeds and results in Fig. 13.

In general, such a transformation can cause issues with learning good state-based critics due the fundamentally non-Markovian nature of a trajectory-level reward. However, we find that in practice, this reward transformation (Line 7 in Algorithm 2) significantly speeds up policy search.

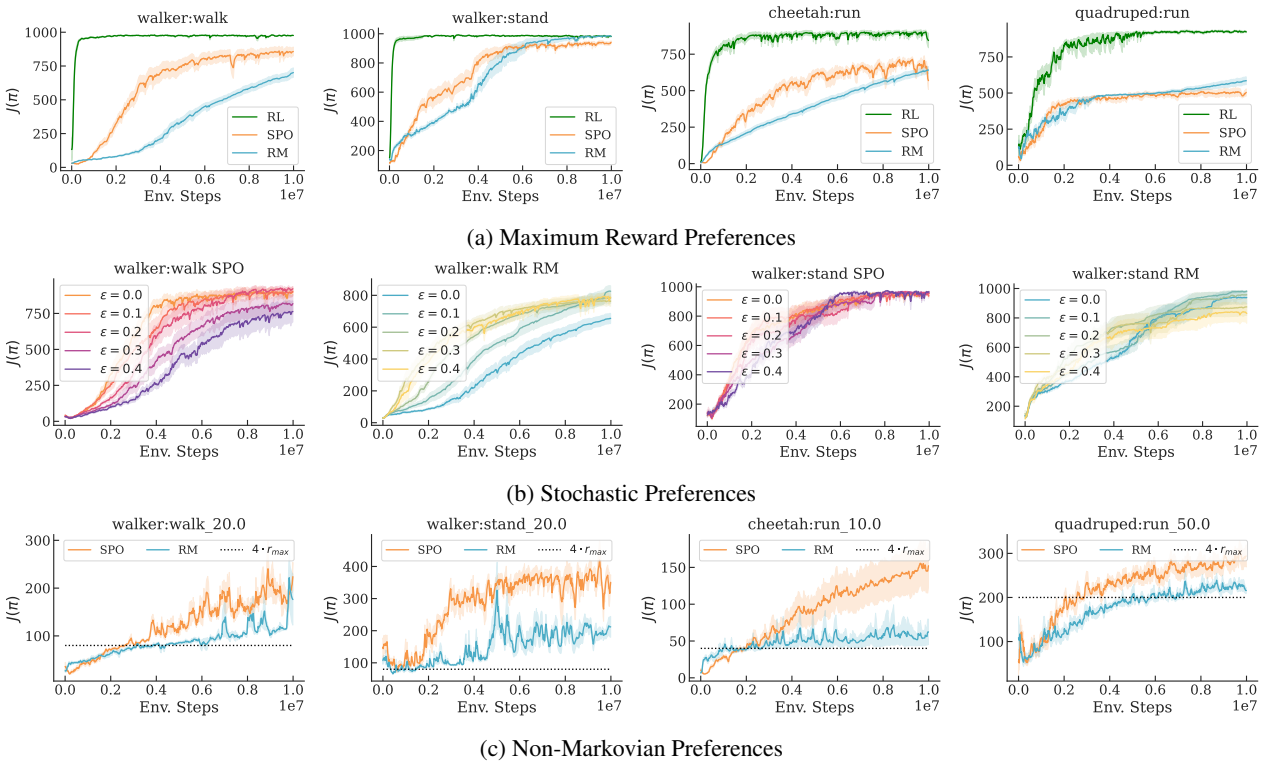
Lastly, in theory, SPO requires sampling multiple trajectories per policy update. In practice, we simply keep a queue $\mathcal{Q}$ of a small, fixed size (typically 10-100) and use the win rate against this queue for labeling trajectories sampled from the current policy (Line 6 in Algorithm 2). This makes our approach strikingly lightweight to implement on top of any policy optimization method of choice: *it is just reward relabeling*. We describe our full approach in Algorithm 2. We remark that this trick of maintaining a queue only works in settings without context, and in contextual settings like LLMs, we still need to query and compare multiple actions at each context – see Algorithm 4 for details.

3. Experimental Evaluation

We evaluate SPO experimentally and compare it against an iterative Reward Modeling (RM) approach along several axes. We focus on the context-free, online oracle setting and leave exploration of the contextual, offline dataset setting to future work. Specifically, we ask the following questions:

1. Can SPO compute MWs when faced with intransitive preferences? We consider aggregating three populations in different proportions, each of which has transitive preferences internally. We measure how far off SPO is from the exact MW. We also present qualitative results on a continuous control task from Mujoco, (Brockman et al., 2016) where computing the MW for comparison is infeasible.

2. How sample efficient is SPO on problems with unique



(a) Maximum Reward Preferences

(b) Stochastic Preferences

(c) Non-Markovian Preferences

Figure 6: **(a):** When the preferred trajectory is that with the higher ground-truth reward, SPO is able to match or improve upon the RM-based approach. **(b)** We ablate the robustness of SPO and reward-model based approaches to preference feedback being flipped w.p. ϵ. Despite the fact that SPO doesn't learn an extra model to average out the noisy feedback, on some environments, it is able to match the performance of RM. **(c)** We compare SPO and RM-based approaches on a non-Markovian task: maximize the reward subject to the constraint that the total reward accrued during the last quarter must be at most $r_{\max}$. SPO learns policies that exploit their freedom early in the episode while RM struggles to cross the $4 \cdot r_{\max}$ threshold. Standard errors are computed across 10 seeds.

Copeland Winners / optimal policies? We evaluate SPO and RM with preferences based on ground truth rewards from the DMControl (Tassa et al., 2018) continuous control environments. This setting is tailor-made for RM as there exists a deterministic, Markovian reward function that explains the observed preferences.

3. How robust is SPO to stochastic preferences? We study the robustness of RM and SPO to corruptions of various probabilities (i.e. Bernoulli noise) in preference labels. This setup is meant to capture some of the stochasticity in human preferences that makes RLHF challenging in practice.

4. Can SPO handle Non-Markovian preferences? We consider a challenging situation where we want to elicit *qualitatively* non-Markovian behavior (e.g. constraints on just a part of a trajectory) from a Markovian policy purely on the basis of trajectory-level relative feedback.

To eliminate data staleness issues, we allow both SPO and RM to query the preference function online, and thus continuously update the RM during policy search. Hence, RM can be seen as a *maximally iterative* reward-based method and therefore much closer to SPO than the traditional two-stage procedure often employed in RLHF. [6] As we discussed above, reward-based methods with sufficiently high update frequencies resemble SPO, with the only difference potentially being the training distribution for the underlying RM, when compared with the implicit SPO reward (7). Thus, our investigation focuses on the subtle questions: *whether learning a reward model provides any benefit over directly using preference data when teacher queries are cheap.*

We use Soft Actor Critic (SAC, Haarnoja et al. (2018)) for continuous control and Proximal Policy Optimization (PPO, Schulman et al. (2017)) for discrete action tasks, both as implemented in the ACME framework (Hoffman et al., 2020). The actor, critic sizes and activations are kept same between SPO and RM. Explicitly, the *only* distinction between SPO and RM is how the reward of a trajectory is estimated and fed into the policy optimization algorithm.

The reward model is Markovian and trained on trajectory level comparisons using trajectories drawn from the agent's

[6] In Figure 15, we observe that freezing the reward model partway through training (i.e. moving us closer to an offline RLHF recipe) leads to a consistent decline in performance for RM.

replay buffer by optimizing the Bradley-Terry loss (Bradley & Terry, 1952). [7] During learning, the current reward model is used to label samples that are drawn from the replay buffer for performing policy search. The relabeling works better in our experiments than using the rewards assigned when these samples were put into the replay buffer. We keep the reward model architectures consistent with Lee et al. (2021a) and perform extensive sweeps over learning rates and update rules – see Appendix F for more details. **We postpone additional results for all experiments to Appendix E.**

Intransitive Preferences. We begin by testing whether SPO is actually able to compute MWs in practice via attempting to optimize *cyclic* (intransitive) preferences.

First, we consider 3 populations, each of which has internally transitive preferences over a discrete set of 3 options. However, when aggregated, their preferences become intransitive, as is common in real-world scenarios (May, 1954). Because this problem is discrete, we can compute the MW in closed form. As we show in Fig. 5 (a) (also Figs. 11, 12), SPO almost exactly computes the MW across a range of sub-population weightings, agreeing with Theorem 2.3. In contrast, RM forces a total order over the actions, and the policy typically converges to a corner far from the MW.

Next, we consider the Mujoco Ant-v3 navigating atop a 2D plane and consider its radius and angle with respect to the origin at the end of the episode, $(R(\xi), \theta(\xi))$. We consider a preference where a trajectory looses to the "pizza-slice" of angle $\bar{\theta} = \pi/4$ in front of it and, within each slice, prefers the "crust" to the "cheese". While we are unable to compute the exact MW here, the symmetry of the preference function over angles implies that the MW qualitatively chooses points uniformly across the slices, while keeping a minimum distance from the center. In Figures 5 (b) and 13, we see that over the course of training, our agent matches this behavior on average, continuously sweeping out full circles.

Maximum Reward Preference. We next consider tasks where the preferences are determined according to comparisons based on the underlying ground truth reward function, i.e. $\mathcal{P}_1(\xi, \xi') = 2 \cdot \mathbb{1}[r(\xi) > r(\xi')] - 1$, where $r(\xi) = \sum_{t=1}^{H} r_{\text{GT}}(s_t^\xi, a_t^\xi)$, with $r_{\text{GT}}(s_t^\xi, a_t^\xi)$ denoting the ground truth reward for the t^{th} state-action pair in trajectory ξ. Naturally, this is the most favorable setting for utilizing an RM based approach, where one might hope to leverage the generalization ability of an RM to perform careful credit assignment for sample efficient learning. However, based on Figure 5(b) and 8, this isn't always the case. This dovetails with our Corollary 2.5 on the fast rates enjoyed by SPO when preferences are explained by a reward function.

Noisy Preferences. A natural followup to ground truth reward based preferences is to test the robustness of these methods to noisy preferences (representing annotator disagreements). We study this setting by flipping the maximum reward preference $\mathcal{P}_1(\cdot)$ above according to i.i.d Bernoulli noise k, i.e. $\mathcal{P}_2(\xi, \xi') = (1-k)\mathcal{P}_1(\xi, \xi') + k \cdot \mathcal{P}_1(\xi', \xi)$, where $k \sim \text{Bern}(\epsilon)$. As we do not learn an extra model in our implementation of SPO, we would expect it to be less robust to this sort of noise due to the lack of an additional averaging effect. In Figure 6 (a) and 9, we see that SPO performs comparably to RM on some environments while performing noticeably worse on some. Taken with the preceding result, this result implies that while learning a parametric model from preference feedback has limited utility when raters are always correct, doing so can provide a strong emperical benefit in some stochastic situations. Thus, an interesting question to explore in future work is whether learning an iterative *preference* model helps close this gap.

Non-Markovian Preferences. Lastly, we consider a challenging task where we want the agent to maximize their cumulative reward as much as possible subject to the constraint that their total reward in the last quarter of a trajectory is below a threshold r_{max}. The optimal strategy for this setting is to exhibit *qualitatively* non-Markovian behavior, i.e. maximize reward as much as possible during the first $\frac{3}{4}$ths of an episode before switching to more conservative behavior. The reason this is challenging is because we are optimizing over *Markovian* policies, which means the agent needs to learn an unusually complex mapping. In Figs. 6 (b) and 9, we see that while SPO is consistently able to cross the $4 \cdot r_{\text{max}}$ threshold that corresponds to exhibiting qualitatively non-Markovian behavior, RM often struggles to do so.

4. Discussion

This paper develops SPO, a simple and effective approach to RLHF. We confirm the efficacy both in our theoretical analysis and diverse evaluation in control tasks. In theory, better understanding of last iterate convergence issues with bandit feedback, and handling uncertainty in learned preference functions are interesting research directions. A natural empirical question is applying these ideas to fine-tuning generative models with AI feedback from large models, or using preference models learned from human annotations. Relatedly, an computational limitation of preference model-based methods compared to reward model-based in the contextual setting worth exploring is the need to sample multiple times (at least twice) to compute scores for a generation.

[7] As prior work often uses snippet-level feedback in this domain (Lee et al., 2021a), we ablate its effect in Figure 14 and find that it can either help or hurt depending on the environment. We also do not include several other techniques from Lee et al. (2021a) like reward model ensembles, active query selection, or batched rather than interleaved updates of the reward model as our work focuses on the fundamental differences between preference-based and reward-based methods, rather than query efficiency in RLHF. We note that the aforementioned techniques are equally applicable to both learned reward and preference models.

Acknowledgments

ZSW is supported in part by the NSF FAI Award #1939606, a Google Faculty Research Award, a J.P. Morgan Faculty Award, a Facebook Research Award, an Okawa Foundation Research Grant, and a Mozilla Research Grant. GKS would like to thank Drew Bagnell for valuable feedback.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Afsar, M. M., Crump, T., and Far, B. Reinforcement learning based recommender systems: A survey. *ACM Computing Surveys*, 55(7):1–38, 2022.

Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *The Journal of Machine Learning Research*, 22(1):4431–4506, 2021.

Agranov, M. and Ortoleva, P. Stochastic choice and preferences for randomization. *Journal of Political Economy*, 125(1):40–68, 2017.

Ailon, N. and Mohri, M. Preference-based learning to rank. *Machine Learning*, 80:189–211, 2010.

Akrour, R., Schoenauer, M., and Sebag, M. April: Active preference learning-based reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II 23*, pp. 116–131. Springer, 2012.

Arrow, K. J. A difficulty in the concept of social welfare. *Journal of political economy*, 58(4):328–346, 1950.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

Azar, M. G., Rowland, M., Piot, B., Guo, D., Calandriello, D., Valko, M., and Munos, R. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*, 2023.

Bagnell, J. A., Kakade, S., Ng, A. Y., and Schneider, J. Policy search by dynamic programming. *Advances in neural information processing systems*, 16, 2003.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022a.

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., Chen, C., Olsson, C., Olah, C., Hernandez, D., Drain, D., Ganguli, D., Li, D., Tran-Johnson, E., Perez, E., Kerr, J., Mueller, J., Ladish, J., Landau, J., Ndousse, K., Lukosuite, K., Lovitt, L., Sellitto, M., Elhage, N., Schiefer, N., Mercado, N., DasSarma, N., Lasenby, R., Larson, R., Ringer, S., Johnston, S., Kravec, S., Showk, S. E., Fort, S., Lanham, T., Telleen-Lawton, T., Conerly, T., Henighan, T., Hume, T., Bowman, S. R., Hatfield-Dodds, Z., Mann, B., Amodei, D., Joseph, N., McCandlish, S., Brown, T., and Kaplan, J. Constitutional ai: Harmlessness from ai feedback, 2022b.

Bengs, V., Busa-Fekete, R., Mesaoudi-Paul, A. E., and HÃllermeier, E. Preference-based online learning with dueling bandits: A survey. *Journal of Machine Learning Research*, 22(7):1–108, 2021. URL http://jmlr.org/papers/v22/18-546.html.

Bıyık, E., Huynh, N., Kochenderfer, M. J., and Sadigh, D. Active preference-based gaussian process regression for reward learning. *arXiv preprint arXiv:2005.02575*, 2020.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Brandl, F., Brandt, F., and Seedig, H. G. Consistent probabilistic social choice. *Econometrica*, 84(5):1839–1880, 2016.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *CoRR*, abs/1606.01540, 2016. URL http://arxiv.org/abs/1606.01540.

Cai, Y., Luo, H., Wei, C.-Y., and Zheng, W. Uncoupled and convergent learning in two-player zero-sum markov games with bandit feedback, 2023.

Cakmak, M., Srinivasa, S. S., Lee, M. K., Forlizzi, J., and Kiesler, S. Human preferences for robot-human hand-over configurations. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1986–1993. IEEE, 2011.

Calandriello, D., Guo, D., Munos, R., Rowland, M., Tang, Y., Pires, B. A., Richemond, P. H., Lan, C. L., Valko, M., Liu, T., et al. Human alignment of large language models through online preference optimisation. *arXiv preprint arXiv:2403.08635*, 2024.

Chen, Z., Deng, Y., Yuan, H., Ji, K., and Gu, Q. Self-play fine-tuning converts weak language models to strong language models, 2024.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Daskalakis, C., Ilyas, A., Syrgkanis, V., and Zeng, H. Training gans with optimism. *arXiv preprint arXiv:1711.00141*, 2017.

Daskalakis, C., Foster, D. J., and Golowich, N. Independent policy gradient methods for competitive reinforcement learning. *Advances in neural information processing systems*, 33:5527–5540, 2020.

De Gemmis, M., Iaquinta, L., Lops, P., Musto, C., Narducci, F., and Semeraro, G. Preference learning in recommender systems. *Preference Learning*, 41:41–55, 2009.

Dudík, M., Hofmann, K., Schapire, R. E., Slivkins, A., and Zoghi, M. Contextual dueling bandits. In *Conference on Learning Theory*, pp. 563–587. PMLR, 2015.

Even-Dar, E., Kakade, S. M., and Mansour, Y. Online markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.

Fey, M. Symmetric games with only asymmetric equilibria. *Games and Economic Behavior*, 75(1):424–427, 2012.

Fishburn, P. C. Probabilistic social choice based on simple voting comparisons. *The Review of Economic Studies*, 51 (4):683–692, 1984.

Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

Gao, Z., Chang, J. D., Zhan, W., Oertell, O., Swamy, G., Brantley, K., Joachims, T., Bagnell, J. A., Lee, J. D., and Sun, W. Rebel: Reinforcement learning via regressing relative rewards. *arXiv preprint arXiv:2404.16767*, 2024.

Gardner, M. Mathematical games, Dec 1970. URL https://www.scientificamerican.com/article/mathematical-games-1970-12/.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Hazan, E. et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Hejna, J., Rafailov, R., Sikchi, H., Finn, C., Niekum, S., Knox, W. B., and Sadigh, D. Contrastive prefence learning: Learning from human feedback without rl. *arXiv preprint arXiv:2310.13639*, 2023.

Hoffman, M. W., Shahriari, B., Aslanides, J., Barth-Maron, G., Momchev, N., Sinopalnikov, D., Stańczyk, P., Ramos, S., Raichuk, A., Vincent, D., et al. Acme: A research framework for distributed reinforcement learning. *arXiv preprint arXiv:2006.00979*, 2020.

Ibarz, B., Leike, J., Pohlen, T., Irving, G., Legg, S., and Amodei, D. Reward learning from human preferences and demonstrations in atari. *Advances in neural information processing systems*, 31, 2018.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 267–274, 2002.

Kakade, S. M. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

Kalai, A. and Vempala, S. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Kramer, G. H. On a class of equilibrium conditions for majority rule. *Econometrica*, 41(2):285–97, 1973. URL https://EconPapers.repec.org/RePEc:ecm:emetrp:v:41:y:1973:i:2:p:285-97.

Kreweras, G. Aggregation of preference orderings. In *Mathematics and Social Sciences I: Proceedings of the seminars of Menthon-Saint-Bernard, France (1–27 July 1960) and of Gösing, Austria (3–27 July 1962)*, pp. 73–79, 1965.

Lee, K., Smith, L., and Abbeel, P. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. *arXiv preprint arXiv:2106.05091*, 2021a.

Lee, K., Smith, L., Dragan, A., and Abbeel, P. B-pref: Benchmarking preference-based reinforcement learning. *arXiv preprint arXiv:2111.03026*, 2021b.

Leslie, D. S. and Collins, E. J. Generalised weakened fictitious play. *Games and Economic Behavior*, 56(2):285–298, 2006.

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

McMahan, B. Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 525–533. JMLR Workshop and Conference Proceedings, 2011.

Miller, G. A. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

Munos, R., Valko, M., Calandriello, D., Azar, M. G., Rowland, M., Guo, Z. D., Tang, Y., Geist, M., Mesnard, T., Michi, A., Selvi, M., Girgin, S., Momchev, N., Bachem, O., Mankowitz, D. J., Precup, D., and Piot, B. Nash learning from human feedback, 2023.

Nash, J. Non-cooperative games. *Annals of mathematics*, pp. 286–295, 1951.

Novoseller, E., Wei, Y., Sui, Y., Yue, Y., and Burdick, J. Dueling posterior sampling for preference-based reinforcement learning. In *Conference on Uncertainty in Artificial Intelligence*, pp. 1029–1038. PMLR, 2020.

OpenAI, :, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk, Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Michael, Pokorny, Pokrass, M., Pong, V., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. Gpt-4 technical report, 2023.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Pacchiano, A., Saha, A., and Lee, J. Dueling rl: Reinforcement learning with trajectory preferences, 2023.

Pomerleau, D. A. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

Puterman, M. L. *Markov decision processes: discrete*

stochastic dynamic programming. John Wiley & Sons, 2014.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Rosset, C., Cheng, C.-A., Mitra, A., Santacroce, M., Awadallah, A., and Xie, T. Direct nash optimization: Teaching language models to self-improve with general preferences. *arXiv preprint arXiv:2404.03715*, 2024.

Sadigh, D., Dragan, A. D., Sastry, S., and Seshia, S. A. Active preference-based learning of reward functions. 2017.

Saha, A. and Krishnamurthy, A. Efficient and optimal algorithms for contextual dueling bandits under realizability. In *International Conference on Algorithmic Learning Theory*, pp. 968–994. PMLR, 2022.

Saha, A., Koren, T., and Mansour, Y. Adversarial dueling bandits. In *International Conference on Machine Learning*, pp. 9235–9244. PMLR, 2021.

Satterthwaite, M. A. Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of economic theory*, 10(2):187–217, 1975.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sen, A. Social choice theory. *Handbook of mathematical economics*, 3:1073–1181, 1986.

Sikchi, H., Saran, A., Goo, W., and Niekum, S. A ranking game for imitation learning. *arXiv preprint arXiv:2202.03481*, 2022.

Simpson, P. B. On Defining Areas of Voter Choice: Professor Tullock on Stable Voting. *The Quarterly Journal of Economics*, 83(3):478–490, 08 1969. ISSN 0033-5533. doi:10.2307/1880533. URL https://doi.org/10.2307/1880533.

Sion, M. On general minimax theorems. 1958.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33: 3008–3021, 2020.

Sui, Y., Zhuang, V., Burdick, J. W., and Yue, Y. Multi-dueling bandits with dependent arms. *arXiv preprint arXiv:1705.00253*, 2017.

Swamy, G., Reddy, S., Levine, S., and Dragan, A. D. Scaled autonomy: Enabling human operators to control robot fleets. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5942–5948. IEEE, 2020.

Swamy, G., Choudhury, S., Bagnell, J. A., and Wu, S. Of moments and matching: A game-theoretic framework for closing the imitation gap. In *International Conference on Machine Learning*, pp. 10022–10032. PMLR, 2021.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model, 2023.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., de Las Casas, D., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., Lillicrap, T. P., and Riedmiller, M. A. Deepmind control suite. *CoRR*, abs/1801.00690, 2018. URL http://arxiv.org/abs/1801.00690.

Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models, 2023.

Tucker, M., Novoseller, E., Kann, C., Sui, Y., Yue, Y., Burdick, J. W., and Ames, A. D. Preference-based learning

for exoskeleton gait optimization. In *2020 IEEE international conference on robotics and automation (ICRA)*, pp. 2351–2357. IEEE, 2020.

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

Viappiani, P. and Boutilier, C. Optimal bayesian recommendation sets and myopically optimal choice query sets. *Advances in neural information processing systems*, 23, 2010.

Wang, Y., Liu, Q., and Jin, C. Is rlhf more difficult than standard rl?, 2023.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

Wirth, C., Akrour, R., Neumann, G., Fürnkranz, J., et al. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136): 1–46, 2017.

Yue, Y. and Joachims, T. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1201–1208, 2009.

Yue, Y., Broder, J., Kleinberg, R., and Joachims, T. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

Zhao, Y., Joshi, R., Liu, T., Khalman, M., Saleh, M., and Liu, P. J. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*, 2023.

Zhu, B., Frick, E., Wu, T., Zhu, H., and Jiao, J. Starling-7b: Improving llm helpfulness & harmlessness with rlaif, November 2023.

Ziebart, B. D. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences, 2020.

Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pp. 928–936, 2003.

Zinkevich, M., Johanson, M., Bowling, M., and Piccione, C. Regret minimization in games with incomplete information. *Advances in neural information processing systems*, 20, 2007.

Zucker, M., Ratliff, N., Stolle, M., Chestnutt, J., Bagnell, J. A., Atkeson, C. G., and Kuffner, J. Optimization and learning for rough terrain legged locomotion. *The International Journal of Robotics Research*, 30(2):175–191, 2011.

A. Related Work

Dueling Bandits and Dueling RL. Beginning with the seminal work of Yue et al. (2012), various authors have viewed preference-based optimization of a multi-armed or contextual bandit as a two-player zero-sum game (Dudík et al., 2015; Saha et al., 2021; Saha & Krishnamurthy, 2022; Bengs et al., 2021). Dudík et al. (2015) carry out a detailed theoretical study of the two-player approach in contextual bandits, where they use the name *von Neumann winner* to refer to the concept of a Minimax Winner. We adopt the latter name due to its older roots in the social choice theory literature. More recently, various authors have investigated dueling reinforcement learning algorithms from a theoretical perspective. In contrast to Pacchiano et al. (2023), we do not need to assume preferences are explained by an underlying reward function. We build upon the work of Wang et al. (2023) by utilizing their reduction of RLHF to adversarial MDP solving. However, we further leverage the structure of the problem to derive single-player algorithms, while all aforementioned approaches requires adversarial training.

Both in the bandit and sequential settings, prior work has considered single-player algorithms for RLHF. However, these results require strong linearity assumptions and are only applicable in the bandit domain (Sui et al., 2017) or assume an underlying Markovian reward function (Novoseller et al., 2020). In contrast, we provide a reduction to no-regret online learning that allows one to plug in *any* no-regret algorithm (e.g. Online Gradient Descent, Zinkevich (2003)) without additional assumptions. In particular, by leveraging a recent result from Wang et al. (2023), we are able to prove that we can utilize a variation of the Natural Policy Gradient (Kakade, 2001; Agarwal et al., 2021), which practical policy gradient algorithms like PPO (Schulman et al., 2017) and TRPO (Schulman et al., 2015) approximate, to efficiently compute Minimax Winners, bypassing the hardness result of Daskalakis et al. (2020) by utilizing the structure of our particular game.

Perhaps the most similar work to ours is the concurrent work of Munos et al. (2023). They derive a specific algorithm, focus on quantal response equilibria to be able to prove last-iterate convergence, and treat the problem as a normal-form game. Instead, we focus on a general algorithmic framework, provide convergence guarantees to the Nash equilibrium, and account for the sequential nature of the game. Empirically, they focus on a particular task of learning document summarization from human feedback, while we study continuous control tasks where we experiment with a range of realistic preference functions. Recent work by Chen et al. (2024) formulates inverse RL for LLM fine-tuning as a kind of self-play – we focus on optimizing from preferences rather than from demonstrations.

Our reward model baseline and our continuous control setup are heavily influenced by the works of Christiano et al. (2017); Lee et al. (2021a). One critical distinction from this prior work is rather than assuming that the rater is able to provide snippet-level feedback, we only assume they can perform trajectory-level comparisons, a much less dense form of supervision.

RLHF without Reward Models. Recently, several authors have proposed eliminating reward models from RLHF by leveraging the well-known bijection between the optimal policies of minimum-relative-entropy RL problems and their advantage functions (Ziebart, 2010) to directly optimize the policy by substituting it into the classification loss usually used to train the reward model (Zhao et al., 2023; Rafailov et al., 2023; Hejna et al., 2023; Azar et al., 2023). While these approaches are appealing for their conceptual simplicity and ease of implementation, they suffer from the same issues with intransitive and noisy preferences as they are derived on the basis of an *implicit* reward model. Additionally, as we discuss further in Appendix D, they can suffer from *compounding errors* (Ross et al., 2011) due to their offline nature. In a sense, this family of approaches can be thought of as the preference-based analog to *behavioral cloning* techniques (Pomerleau, 1988) in imitation learning. [8] In contrast, the technique we propose is the preference-based analog of inverse reinforcement learning approaches (Ziebart, 2010). We summarize this taxonomy in Table 1.

Self-Play In Language Modeling. Various works have explored self-play for preference fine-tuning LLMs. Conceptually, this can be thought of as an instantiation of the general `SPO` reduction on a contextual bandit problem, using a variety of base learners to approximate the idealized update we propose. Concurrent to our work, Munos et al. (2023) propose using policy gradients to optimize a family of objectives, including the `SPO` losses. Subsequent to our work, Calandriello et al. (2024) propose using Online IPO, Rosset et al. (2024) propose using Online DPO, and Gao et al. (2024) propose using Online REBEL to optimize the sequence of `SPO` losses. We do not explore the contextual nor preference fine-tuning settings in our experiments and refer interested readers to the above papers for details on scalable approximations of `SPO`.

[8] While preferences are collected on the initial policy's distribution (unlike in BC), this data becomes off-policy after policy optimization.

B. Frequency of Querying Online Preference Oracle.

Our presentation assumed access to a preference function $\mathcal{P}$ which can be queried at each round with new trajectories. This is natural when preference labels are generated by a model learned from a previously collected preference dataset, as in recently studied RLAIF settings (Bai et al., 2022a; Zhao et al., 2023). However, in other settings, it is desirable to directly obtain the preference labels from humans or other expert models (RLHF, Tucker et al. (2020)), where online querying is not possible. SPO is compatible with *batched queries* the queries in these settings. Specifically, we can freeze the policy for some batch size B, accumulate a dataset of B trajectory pairs to compare, and then query the labels for all of them. This corresponds to mini-batching in the no-regret algorithm being used by SPO, which preserves no-regret for $B < O(\sqrt{T})$.

C. Proofs

Contents

C.1. Proof of Lemma 2.1

Proof. We follow a similar proof strategy to Fey (2012). For convenience, we interpret the preference $\mathcal{P}$ as a matrix of size $|\Pi| \times |\Pi|$ and will write things in matrix notation, i.e. $\mathbb{E}_{\pi_1 \sim p, \pi_2 \sim q}[\mathcal{P}(\pi_1, \pi_2)] = p^T \mathcal{P} q$. We consider some $(\hat{p}, \hat{q}) \in \mathsf{MW}(\mathcal{P})$ and will show that also $(\hat{q}, \hat{p}) \in \mathsf{MW}(\mathcal{P})$. Since $(\hat{p}, \hat{q}) \in \mathsf{MW}(\mathcal{P})$ and by the definition of a Nash equilibrium,

$$
\begin{aligned}
&\max_{q \in \Delta(\Pi)} \hat{p}^T \mathcal{P} q \leq \min_{p \in \Delta(\Pi)} p^T \mathcal{P} \hat{q} \\
\Rightarrow &\max_{q \in \Delta(\Pi)} q^T \mathcal{P}^T \hat{p} \leq \min_{p \in \Delta(\Pi)} \hat{q}^T \mathcal{P}^T p \\
\Rightarrow &\max_{q \in \Delta(\Pi)} -q^T \mathcal{P} \hat{p} \leq \min_{p \in \Delta(\Pi)} -\hat{q}^T \mathcal{P} p && \text{(By the anti-symmetry of } \mathcal{P}) \\
\Rightarrow &- \min_{q \in \Delta(\Pi)} q^T \mathcal{P} \hat{p} \leq - \max_{p \in \Delta(\Pi)} \hat{q}^T \mathcal{P} p \\
\Rightarrow &\min_{q \in \Delta(\Pi)} q^T \mathcal{P} \hat{p} \geq \max_{p \in \Delta(\Pi)} \hat{q}^T \mathcal{P} p \\
\Rightarrow &\max_{q \in \Delta(\Pi)} \hat{q}^T \mathcal{P} q \leq \min_{p \in \Delta(\Pi)} p^T \mathcal{P} \hat{p}
\end{aligned}
$$

Thus, $(\hat{q}, \hat{p})$ also forms a Nash equilibrium. Then, because of the interchangeability of Nash equilibrium strategies for two-player zero-sum games (Nash, 1951), we have that $(\hat{p}, \hat{p})$ and $(\hat{q}, \hat{q})$ are symmetric MWs. □

C.2. Proof of Theorem 2.3

Proof. We follow the strategy outlined in our preceding proof sketch.

Consider two players, $p, q \in \Delta(\Pi)$. An ϵ-approximate Nash equilibrium is a pair of strategies (p, q) such that

$$
\max_{p^\star \in \Delta(\Pi)} \mathbb{E}_{\pi \sim p^\star, \pi' \sim q}[\mathcal{P}(\pi, \pi'))] - \min_{q^\star \in \Delta(\Pi)} \mathbb{E}_{\pi \sim p, \pi' \sim q^\star}[\mathcal{P}(\pi, \pi'))] \leq \epsilon. \tag{8}
$$

We define the following per-round losses for both players:

$$
\ell_t^1(p) = \mathbb{E}_{\pi \sim p, \pi' \sim q_t}[-\mathcal{P}(\pi, \pi')], \quad \ell_t^2(q) = \mathbb{E}_{\pi \sim p_t, \pi' \sim q}[\mathcal{P}(\pi, \pi')]. \tag{9}
$$

We can then define the (static) regret suffered by both players as

$$\mathsf{Reg}_p(T) = \sum_t^T \ell_t^1(p_t) - \min_{p^\star \in \Delta(\Pi)} \sum_t^T \ell_t^1(p^\star), \quad \mathsf{Reg}_q(T) = \sum_t^T \ell_t^2(q_t) - \min_{q^\star \in \Delta(\Pi)} \sum_t^T \ell_t^2(q^\star). \tag{10}$$

By construction, we set $p_0 = q_0$. This implies that

$$\begin{aligned}
\ell_0^1(\sigma) &= \mathbb{E}_{\pi\sim\sigma,\pi'\sim q_0}[-\mathcal{P}(\pi,\pi')] && (11)\\
&= \mathbb{E}_{\pi\sim\sigma,\pi'\sim p_0}[-\mathcal{P}(\pi,\pi')] && (12)\\
&= \mathbb{E}_{\pi\sim p_0,\pi'\sim\sigma}[\mathcal{P}(\pi,\pi')] && \text{(By the anti-symmetry of } \mathcal{P}) \\
&= \ell_0^2(\sigma) && (13)
\end{aligned}$$

If, at some time τ, p_τ and q_τ have the same strategy and perform updates based on the same loss function, $p_{\tau+1} = q_{\tau+1}$.[9] Then, by the anti-symmetry of $\mathcal{P}$, we have that $\ell_{\tau+1}^1 = \ell_{\tau+1}^2$. Therefore, by induction, we have that $p_t = q_t$, $\forall t \in [T]$.

We complete the proof by following the argument in Freund & Schapire (1997):

$$\begin{aligned}
\frac{\mathsf{Reg}_p(T) + \mathsf{Reg}_q(T)}{T} &= \frac{1}{T}\left(\sum_t^T \ell_t^1(p_t) - \min_{p^\star\in\Delta(\Pi)} \sum_t^T \ell_t^1(p^\star) + \sum_t^T \ell_t^2(q_t) - \min_{q^\star\in\Delta(\Pi)} \sum_t^T \ell_t^2(q^\star)\right) && (14)\\
&= \frac{1}{T}\left(-\min_{p^\star\in\Delta(\Pi)} \sum_t^T \ell_t^1(p^\star) - \min_{q^\star\in\Delta(\Pi)} \sum_t^T \ell_t^2(q^\star)\right) && (\ell_t^1(p_t) + \ell_t^2(q_t) = 0)\\
&= \frac{1}{T}\left(-\min_{p^\star\in\Delta(\Pi)} \sum_t^T \mathbb{E}_{\pi\sim p^\star,\pi'\sim q_t}[-\mathcal{P}(\pi,\pi'))] - \min_{q^\star\in\Delta(\Pi)} \sum_t^T \mathbb{E}_{\pi\sim p_t,\pi'\sim q^\star}[\mathcal{P}(\pi,\pi'))]\right) && (15)\\
&= \left(\max_{p^\star\in\Delta(\Pi)} \mathbb{E}_{\pi\sim p^\star,\pi'\sim\overline{q}}[\mathcal{P}(\pi,\pi'))] - \min_{q^\star\in\Delta(\Pi)} \mathbb{E}_{\pi\sim\overline{p},\pi'\sim q^\star}[\mathcal{P}(\pi,\pi'))]\right). && (16)
\end{aligned}$$

Thus, $(\overline{p},\overline{q}) = (\overline{p},\overline{p})$ is a symmetric $\frac{\mathsf{Reg}_p(T)+\mathsf{Reg}_q(T)}{T} = \frac{2\mathsf{Reg}_p(T)}{T}$-approximate Nash equilibrium / Minimax Winner. □

C.3. Proof of Theorem 2.4

Proof. Our preceding proof sketch ignored the effect of regularization for simplicity. We now provide a specific example under which standard algorithms do not compute Minimax Winners, even with regularization to a prior.

We set π_{ref} to be uniform to remove any trivial failures due to a lack of data support and consider the following preference matrix:

	a	b	c
a	0	$\frac{2}{5}$	-1
b	$-\frac{2}{5}$	0	$+1$
c	$+1$	-1	0

Figure 7: A preference function over $\mathcal{Y} = (a, b, c)$ with unique Minimax Winner $(\frac{5}{12}, \frac{5}{12}, \frac{1}{6})$ and unique Copeland winner b.

We begin by considering standard RLHF algorithms. First, we would fit a reward model via the standard Bradley-Terry loss. This would peak at the Copeland Winner, which is b for the above matrix. Without loss of generality, we assume reward model outputs are in the range $[0, 1]$. Thus,

[9] This is trivially true if $\mathcal{O}$ is deterministic. If $\mathcal{O}$ is a randomized no-regret algorithm like Follow the Perturbed Leader (Kalai & Vempala, 2005), we can instead simply share the randomness between the two players.

$$r^\star = [0, 1, 0]. \tag{17}$$

From Ziebart (2010), we have that

$$\pi^\star(y) \propto \pi_{\text{ref}}(y) \exp\left(\frac{1}{\beta} r^\star(y)\right) = \left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right] \odot \left[1, \exp(\frac{1}{\beta}), 1\right] \propto \left[1, \exp(\frac{1}{\beta}), 1\right]. \tag{18}$$

Thus, for any finite non-negative β, $\pi^\star$ plays a and c equally often, which means it cannot play the Minimax Winner.

Next, we consider DPO. From Eq. 6 of Rafailov et al. (2023), we have that the optimal DPO policy has the form

$$p^\star(y_1 \succ y_2) = \frac{1}{1 + \exp\left(\beta \log \frac{\pi^\star(y_2)}{\pi_{\text{ref}}(y_2)} - \beta \log \frac{\pi^\star(y_1)}{\pi_{\text{ref}}(y_1)}\right)}. \tag{19}$$

As π_{ref} is uniform, we have that $\pi_{\text{ref}}(y_2) = \pi_{\text{ref}}(y_1)$ and thus we can simplify our above expression

$$p^\star(y_1 \succ y_2) = \frac{1}{1 + \exp\left(\beta \log \pi^\star(y_2) - \beta \log \pi^\star(y_1)\right)} \tag{20}$$

$$= \frac{1}{1 + \left(\frac{\pi^\star(y_2)}{\pi^\star(y_1)}\right)^\beta} \tag{21}$$

$$= \frac{\pi^\star(y_1)^\beta}{\pi^\star(y_1)^\beta + \pi^\star(y_2)^\beta}. \tag{22}$$

As written, the DPO loss assumes unweighted preferences (i.e. it assumes that all positive samples are equally preferable to their corresponding negative samples). We therefore perform the natural transformation from log likelihood to cross entropy:

$$\ell_{\text{DPO}}(\pi) = -\sum_{y_1, y_2 \in \mathcal{Y} \times \mathcal{Y}} \left[\mathcal{P}(y_1 \succ y_2) \log\left(p^\star(y_1 \succ y_2)\right)\right] \tag{23}$$

$$= -\sum_{y_1, y_2 \in \mathcal{Y} \times \mathcal{Y}} \left[\frac{\mathcal{P}(y_1, y_2) + 1}{2} \log\left(\frac{\pi(y_1)^\beta}{\pi(y_1)^\beta + \pi(y_2)^\beta}\right)\right] \tag{24}$$

$$\tag{25}$$

Via Gibbs' inequality, we know that cross-entropy is minimized when the two distributions are equal. We can then plug in the values from our above preference matrix to arrive at the following set of constraints:

$$\frac{7}{10} = \frac{\pi^\star(a)^\beta}{\pi^\star(a)^\beta + \pi^\star(b)^\beta}, \quad 1 = \frac{\pi^\star(b)^\beta}{\pi^\star(b)^\beta + \pi^\star(c)^\beta}, \quad 1 = \frac{\pi^\star(c)^\beta}{\pi^\star(a)^\beta + \pi^\star(c)^\beta}. \tag{26}$$

Clearly, it is impossible to simultaneously satisfy all of these constraints. Thus, DPO is unable to learn the minimizer of the preference-level cross-entropy loss function because of its assumption of an implicit reward model. Unfortunately, this makes analyzing the solution DPO would actually pick rather difficult from a theoretical perspective. In response, we show that regardless of the setting of β, there exists another strategy (π_{ref}) with a lower loss than the Minimax Winner.

We can write out the above loss more explicitly as

$$\ell_{\text{DPO}}(\pi) = -\left(\frac{7}{10} \log\left(\frac{\pi(a)^\beta}{\pi(a)^\beta + \pi(b)^\beta}\right) + \frac{3}{10} \log\left(\frac{\pi(b)^\beta}{\pi(a)^\beta + \pi(b)^\beta}\right) + \log\left(\frac{\pi(b)^\beta}{\pi(b)^\beta + \pi(c)^\beta}\right) + \log\left(\frac{\pi(c)^\beta}{\pi(a)^\beta + \pi(c)^\beta}\right)\right)$$

First, observe that plugging in π_{ref} gives us a loss value $\log(2)$ for any value of β. Also, observe that $\lim_{\beta \to 0} \ell_{\text{DPO}}(\pi_{\text{MW}}) =$

$\log(2)$. Taking the derivative with respect to β, we get

$$\nabla_\beta \ell_{\text{DPO}}(\pi) = \frac{-7}{10} \log\left(\frac{\pi(a)}{\pi(b)}\right) \frac{\pi(b)^\beta}{\pi(a)^\beta + \pi(b)^\beta} \tag{27}$$

$$+ \frac{-3}{10} \log\left(\frac{\pi(b)}{\pi(a)}\right) \frac{\pi(a)^\beta}{\pi(b)^\beta + \pi(a)^\beta} \tag{28}$$

$$+ -1 \log\left(\frac{\pi(b)}{\pi(c)}\right) \frac{\pi(c)^\beta}{\pi(c)^\beta + \pi(b)^\beta} \tag{29}$$

$$+ -1 \log\left(\frac{\pi(c)}{\pi(a)}\right) \frac{\pi(a)^\beta}{\pi(a)^\beta + \pi(c)^\beta}.$$

Now, plugging in the Minimax Winner, we get that

$$\nabla_\beta \ell_{\text{DPO}}(\pi_{\text{MW}}) = -\log\left(\frac{5}{2}\right) \frac{\pi_{\text{MW}}(c)^\beta}{\pi_{\text{MW}}(c)^\beta + \pi_{\text{MW}}(b)^\beta} - \log\left(\frac{2}{5}\right) \frac{\pi_{\text{MW}}(a)^\beta}{\pi_{\text{MW}}(a)^\beta + \pi_{\text{MW}}(c)^\beta} > 0.$$

Thus, $\forall \beta \in (0, \infty)$, $\ell_{\text{DPO}}(\pi_{\text{MW}}) > \ell_{\text{DPO}}(\pi_{\text{ref}})$. Thus, DPO will never pick the Minimax Winner.

□

C.4. Proof of Corollary 2.6

Proof. We consider optimization over the full history-dependent policy class. That is, for each $\pi \in \Pi$, $\pi = (\pi_1, \ldots, \pi_H)$, where $\pi_h \in \Pi_h = \{\Phi_h \to \Delta(\mathcal{A})\}$. As $\Delta(\mathcal{A})$ is convex and compact, so is Π_h (as the set of functions to a convex set is convex), which means Π is as well (as Cartesian products preserve convexity). We therefore satisfy the conditions for the application of Theorem 2.3. We proceed by bounding the regret of our policy selection strategy. Define $J(\pi, r_t^{\text{SPO}}) = \mathbb{E}_{\xi\sim\pi}[r_t^{\text{SPO}}(\xi)]$ as the performance of the policy π under the induced trajectory-level reward, r_t^{SPO} (Eq. 7). Then,

$$\mathsf{Reg}(T) = \max_{\pi\in\Pi} \sum_{t=1}^{T} \ell_t^{\text{SPO}}(\pi_t) - \ell_t^{\text{SPO}}(\pi) \tag{30}$$

$$= \max_{\pi\in\Pi} \sum_{t=1}^{T} J(\pi, r_t^{\text{SPO}}) - J(\pi_t, r_t^{\text{SPO}}) \tag{31}$$

$$= \max_{\pi\in\Pi} H\mathbb{E}_{\substack{h\sim\mathsf{Unif}([H])\\ \phi\sim\rho_\pi^h}} \left[\sum_{t=1}^{T} Q_t^h(\phi, \pi) - Q_t^h(\phi, \pi_t)\right] \qquad \text{(By the finite-horizon PDL, Bagnell et al. (2003))}$$

In the last step of the proof, we apply Bagnell et al. (2003)'s finite horizon variant of Kakade & Langford (2002)'s Performance Difference Lemma (PDL) to the history-based MDP. The no-regret policy update in Line 9 of the Algorithm 1 is equivalent to running Hedge (Freund & Schapire, 1997) with $\ell_t^{\phi,h}(a) = \frac{1}{2}(1 - Q_t^h(\phi, a)) \in [0, 1]$ as loss function for all $h \in [H]$ and $\phi \in \Phi_h$ (A_h^t and Q_h^T are interchangeable as they differ by a per-state constant and we're taking a softmax). Thus, by the regret bound of Hedge, we have that for any comparator policy $\pi \in \Pi$, $h \in [H]$, and $\phi \in \Phi_h$,

$$\sum_{t=1}^{T} Q_t^h(\phi, \pi) - Q_t^h(\phi, \pi_t) \leq 4\sqrt{\log(|\mathcal{A}|)T}. \tag{32}$$

We can then apply this bound history-wise to our preceding expression to arrive at overall regret bound

$$\mathsf{Reg}(T) \leq 4H\sqrt{\log(|\mathcal{A}|)T}. \tag{33}$$

Thus, by our Theorem 2.3, we have that $\bar{\pi}$ (the trajectory-level mixture of $\pi_{1:T}$) is a $8H\sqrt{\log(|\mathcal{A}|)/T}$-approximate MW.

□

C.5. Proof of Lemma 2.7

Proof. Observe that for any $\xi \in (\mathcal{S} \times \mathcal{A})^H$, $r(\xi) = \sum_{h=1}^H \tilde{r}(s_t, a_t)$. Thus, $\forall \pi \in \Pi$,

$$\mathbb{E}_{\xi\sim\pi}\left[r(\xi)\right] = \mathbb{E}_{\xi\sim\pi}\left[\sum_{h=1}^H \tilde{r}(s_t,a_t)\right] \Rightarrow \operatorname*{argmax}_{\pi\in\Pi} \mathbb{E}_{\xi\sim\pi}\left[r(\xi)\right] = \operatorname*{argmax}_{\pi\in\Pi} \mathbb{E}_{\xi\sim\pi}\left[\sum_{h=1}^H \tilde{r}(s_t,a_t)\right]. \tag{34}$$

□

C.6. Proof of Corollary 2.5

We begin by stating the core assumption we will use in this section.

Assumption C.1 (Gap Condition)**.** There is a subset $\Pi^\star \subseteq \Pi$ such that:

1. $\forall \pi^\star \in \Pi^\star, \pi \in \Pi/\Pi^\star, \mathcal{P}(\pi^\star, \pi) \geq \Delta$.
2. $\forall \pi_1^\star, \pi_2^\star \in \Pi^\star, -\Delta/2 \leq \mathcal{P}(\pi_1^\star, \pi_2^\star) \leq \Delta/2$.
3. $\forall \pi_1, \pi_2 \in \Pi/\Pi^\star, -\Delta \leq \mathcal{P}(\pi_1, \pi_2) \leq \Delta$.

Under this assumption, we can prove that rather than the $\widetilde{O}(\frac{1}{\sqrt{T}})$ rate we usually get for Hedge, we instead get a $\widetilde{O}(\frac{1}{T})$ rate.

Corollary C.2. *Under Assumption C.1, after T calls to Hedge, $\bar{p}$ is an $\frac{1+2|\Pi|\ln T}{\Delta T}$-approximate Minimax Winner.*

Proof. We start by observing that under Assumption C.1, the loss function $\ell_t^1(p) = \mathbb{E}_{\pi\sim p, \pi'\sim q_t}[-\mathcal{P}(\pi, \pi'))]$ observed by the minimizing player satisfies the following two properties:

1. For an policy $\pi \in \Pi^\star$, we have $\ell_t(\pi) \leq \Delta$.
2. For any policy $\pi \in \Pi^\star$ and $\pi' \notin \Pi^\star$, $\ell_t(\pi') \geq \ell_t(\pi) + q_t(\Pi^\star)\Delta/2$, where $q_t(\Pi^\star)$ is the probability of the max player choosing policies from the set $\Pi^\star$.

The second property follows from the fact that the loss of any policy $\pi' \notin \Pi^\star$ is always larger than the loss of any action $\pi^\star \in \Pi^\star$ by Assumption C.1, and the two differ by at least $\Delta/2$ whenever the comparator policy comes from the set $\Pi^\star$, since:

$$\begin{aligned}
\ell_t(\pi') &= \sum_{\pi''\in\Pi} q_t(\pi'')\left[-\mathcal{P}(\pi', \pi'')\right] = \sum_{\pi''\in\Pi^\star} q_t(\pi'')\left[-\mathcal{P}(\pi', \pi'')\right] + \sum_{\pi''\in\Pi/\Pi^\star} q_t(\pi'')\left[-\mathcal{P}(\pi', \pi'')\right] \\
&\geq \sum_{\pi''\in\Pi^\star} q_t(\pi'')\Delta + \sum_{\pi''\in\Pi/\Pi^\star} q_t(\pi'')(-\Delta) && \text{(C.1, (1) and C.1, (3), resp.)} \\
&\geq \sum_{\pi''\in\Pi^\star} q_t(\pi'')\left[\frac{\Delta}{2} - \mathcal{P}(\pi^\star, \pi'')\right] + \sum_{\pi''\in\Pi/\Pi^\star} q_t(\pi'')\left[-\mathcal{P}(\pi^\star, \pi'')\right] && \text{(C.1, (2) and C.1, (1), resp.)} \\
&= \ell_t(\pi^\star) + q_t(\Pi^\star)\Delta/2.
\end{aligned}$$

Now let us consider the class of Follow The Regularized Leader (FTRL, McMahan (2011)) algorithms, which induce probability distributions as

$$p_t = \operatorname*{argmin}_{p\in\Delta(\Pi)} \eta L_{t-1}^T p + R(p), \tag{35}$$

for some convex regularizer R, and where we define $L_t = \sum_{s=1}^t \ell_s$. As p_t minimizes the above expression, we know it satisfies the first-order optimality conditions, which imply that for any other distribution $p \in \Delta(\Pi)$,

$$\langle \nabla R(p_t) + \eta L_{t-1}, p - p_t\rangle \geq 0. \tag{36}$$

To proceed further, we consider coordinate-wise separable regularizers, that is $R(p) = \sum_{i=1}^{|\Pi|} R_i(p_i)$, and use the distribution $p = p_t + \alpha e_{\pi_a} - \alpha e_{\pi_b}$, for actions $\pi_a \in \Pi^\star$ and $\pi_b \notin \Pi^\star$, with $\alpha < \max(\min(p_t(\pi_a), p_t(\pi_b)), \epsilon)$. We further assume that

$p_t(\pi_a) > 0$, which is naturally satisfied by many no-regret strategies that play in the strict interior of the simplex. With these choices, plugging on our preceding setting for p, and dividing both sides by α, we get that

$$\nabla R_i(p_t(\pi_a)) + \eta L_t(\pi_a) - \nabla R_i(p_t(\pi_b)) - \eta L_t(\pi_b) \geq 0.$$

Rearranging terms and combining with our second property, we get that

$$\nabla R_i(p_t(\pi_b)) - \nabla R_i(p_t(\pi_a)) \leq \eta(L_t(\pi_a) - L_t(\pi_b)) \leq -\eta \sum_{s=1}^{t-1} q_s(\Pi^\star)\Delta/2. \tag{37}$$

We now specialize to the case of Hedge, where $R_i(x) = x \ln x$ and $\nabla R_i(x) = \ln x + 1$ but note that a similar argument holds for a wide variety regularizers (i.e. other no-regret algorithms) under analogous assumptions. Then, by simplifying the RHS and LHS of our preceding expression and exponentiating both sides, we have that

$$p_t(\pi_b) \leq p_t(\pi_a) \exp\left(\frac{-\eta\Delta}{2} \sum_{s=1}^{t-1} q_s(\Pi^\star)\right).$$

Because the term inside the exponential is always negative and therefore the exponential is at most 1, the above expression implies that $\forall \pi_b \notin \Pi^\star$, $p_t(\pi_b) \leq p_t(\pi_a) \leq p_t(\Pi^\star)$. Thus, via Hölder's inequality, we have that $1 = \sum_{\pi\in\Pi} p_t(\pi) \leq |\Pi| p_t(\Pi^\star) \Rightarrow \frac{1}{|\Pi|} \leq p_t(\Pi^\star) \Rightarrow \exp(-p_t(\Pi^\star)) \leq \exp(\frac{-1}{|\Pi|})$. Since the same holds for q_t by symmetry, we can conclude that

$$p_t(\pi_b) \leq p_t(\pi_a) \exp\left(\frac{-\eta t\Delta}{2|\Pi|}\right).$$

Consequently, once we have $\exp(-\eta t\Delta/(2|\Pi|)) \leq \epsilon$, the probability of any action $b \notin \Pi^\star$ is at most ϵ. Inverting the preceding expression shows this happens in at most $t \leq \frac{2|\Pi|}{\eta\Delta} \ln \frac{1}{\epsilon}$ rounds. Hence,

$$\mathsf{Reg}(T) \leq \epsilon T + \frac{2|\Pi|}{\eta\Delta} \ln \frac{1}{\epsilon}.$$

Choosing $\eta = 1$ and $\epsilon = 1/T$ gives a fast rate. □

We note that the linear dependence on Π is similar to the linear dependence on the number of actions in most gap-dependent bounds for UCB algorithms. We also note that in the bandit feedback setting, Equation 37 only changes in that we have importance-sampled estimates $\hat{L}_t(a) - \hat{L}_t(b)$ instead of the population losses. Adding and subtracting the true means gives us the same quantity as Eq. 37 plus a martingale. This term is $O(\eta\sqrt{t})$. For $\sqrt{t} = O(t\Delta/|\Pi|)$, or equivalently, $t = O((|\Pi|/\Delta)^2)$, we get the same bound with slightly different constants.

C.7. Extension to Bandit Feedback

We now provide an extension of our main result to the bandit feedback setting via the standard importance sampling argument.

Theorem C.3. *Assume Π is finite. Let $\pi_t, \pi_t' \sim p_t$. For any fixed $\alpha \in [0,1]$ and for some $\gamma \in [0,1]$, if we set $\mathcal{O}$ to be the Hedge algorithm of Freund & Schapire (1997) and feed it the sequence of loss functions*

$$\hat{\ell}_t^{SPO}(\pi) = -\alpha \frac{\mathbb{1}[\pi = \pi_t]}{p_t(\pi_t)} \mathcal{P}(\pi_t, \pi_t') + (1-\alpha) \frac{\mathbb{1}[\pi = \pi_t']}{p_t(\pi_t')} \mathcal{P}(\pi_t, \pi_t'),$$

and follow strategy $(1-\gamma)p_t + \frac{\gamma}{|\Pi|}$, $\bar{p}$ is an $\widetilde{O}(\frac{1}{\sqrt{T}})$-approximate Minimax Winner. [10]

Proof. Let $\pi_t \sim p_t$, $\pi_t' \sim q_t$. We observe $\mathcal{P}(\pi_t, \pi_t')$ by querying the preference function.

Define

$$\hat{\ell}_t^1(\pi) = -\frac{\mathbb{1}[\pi = \pi_t]}{p_t(\pi_t)} \mathcal{P}(\pi_t, \pi_t'), \quad \hat{\ell}_t^2(\pi) = \frac{\mathbb{1}[\pi = \pi_t']}{q_t(\pi_t')} \mathcal{P}(\pi_t, \pi_t') \tag{38}$$

[10] For last iterate convergence, one can instead run the no-regret bandit feedback algorithm of Cai et al. (2023) on $\hat{\ell}_t^{\text{SPO}}$, albeit at the cost of a slower rate.

and as the importance-weighted losses for each player at round t. Next, observe that

$$\mathbb{E}_{p_t}[\hat{\ell}_t^1] = \ell_t^1, \mathbb{E}_{q_t}[\hat{\ell}_t^2] = \ell_t^2, \tag{39}$$

i.e. that they are unbiased estimates of the full-feedback losses. We now proceed similarly to our proof for the full feedback case. If we set $p_0 = q_0$, we have that $\ell_0^1 = \ell_0^2$ by the anti-symmetry of $\mathcal{P}$. We also have that

$$\mathbb{E}_{p_0}[\hat{\ell}_0^{\texttt{SPO}}] = \mathbb{E}_{p_0}[\alpha\hat{\ell}_0^1 + (1-\alpha)\hat{\ell}_0^2] = \ell_0^1 = \ell_0^2. \tag{40}$$

We now proceed by induction. If $p_\tau = q_\tau$ at some time τ and we feed both players the same loss $\hat{\ell}_\tau^{\texttt{SPO}}$, by the determinism of Hedge, we have that $p_{\tau+1} = q_{\tau+1}$. This implies $\ell_{\tau+1}^1 = \ell_{\tau+1}^2$, which in turn implies $\hat{\ell}_{\tau+1}^{\texttt{SPO}}$ is an unbiased estimate of both $\ell_{\tau+1}^1$ and $\ell_{\tau+1}^2$. Thus, by induction, we have that both players will have the same iterates and $\hat{\ell}_t^{\texttt{SPO}}$ is an unbiased estimate of both losses $\forall t \in [T]$, which means we can simulate game-solving with a single player.

To prove that above loss estimation scheme preserves the no-regret property, we can simply examine the proof of Exp3 in Auer et al. (2002) and note that the only property required of the loss estimate ($\hat{x}_t$ in the original paper) is that it is unbiased. Thus, we inherit the regret rate of Exp3, which when plugged in completes the proof.

□

We note that while the above is not a general reduction per se, for a wide set of no-regret algorithms, a similar argument applies, with slight differences in the effect of importance sampling on the final regret rate.

C.8. Extension to Contextual Setting

We now extend our above setup to include contexts. Consider a finite-horizon reward-free contextual Markov Decision Process (MDP) (Puterman, 2014) parameterized by $\langle \mathcal{S}, \mathcal{A}, \mathcal{X}, \mathcal{T}, H, \rho \rangle$ where $\mathcal{S}$, $\mathcal{A}$, $\mathcal{X}$ are the state, action, and context spaces, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition operator, H is the horizon, and $\rho : \Delta(\mathcal{X})$ is the context / initial state distribution. We use $\Xi \triangleq (\mathcal{S} \times \mathcal{A})^H$ to denote the space of trajectories and $\Phi_h \triangleq \mathcal{X} \times (\mathcal{S} \times \mathcal{A})^{h-1} \times \mathcal{S}$ to denote the space of h-length histories. We assume that we are given access to a *(contextual) preference function*

$$\mathcal{P} : \mathcal{X} \times \Xi \times \Xi \to [-1, 1] \tag{41}$$

which, given two trajectories $\xi_1, \xi_2 \in \Xi$, outputs a scalar that indicates which is preferred relative to the other. By construction, preference functions are anti-symmetric, i.e. $\forall x, \xi_1, \xi_2 \in \mathcal{X} \times \Xi \times \Xi$, $\mathcal{P}(x, \xi_1, \xi_2) = -\mathcal{P}(x, \xi_2, \xi_1)$. Similarly, we also have that $\forall x, \xi \in \mathcal{X} \times \Xi$, $\mathcal{P}(x, \xi, \xi) = 0$. We assume access to a convex and compact policy class $\Pi \subseteq \{\mathcal{S} \times \mathcal{X} \to \Delta(\mathcal{A})\}$. With a slight abuse of notation, we can define the preference function over policy pairs as

$$\mathcal{P}(\pi_1, \pi_2) \triangleq \mathbb{E}_{x\sim\rho, \xi_1\sim(\pi_1, x), \xi_2\sim(\pi_2, x)}[\mathcal{P}(x, \xi_1, \xi_2)].$$

We now re-state our algorithms, including the dependence on context. Their theoretical guarantees match those presented in the main paper. The main difference in practice is that rather than simply maintaining a queue, we now have to sample multiple trajectories based on a single context for comparison.

Algorithm 3 SPO (Theoretical Version with Contexts)

Input: Learning rate η, Iterations T, Preference fn. $\mathcal{P}$.
Output: Trained policy π.
Initialize $\pi_h(\cdot|\phi) = \mathsf{Unif}(\mathcal{A}), \forall \phi \in \Phi_h, h \in [H]$.
for t in $1 \dots T$ **do**
  Observe context $x_t \sim \rho$.
  Compute $r_t(\xi) = \mathbb{E}_{\xi' \sim \pi_t} \mathcal{P}(x_t, \xi, \xi')$.
  Define $Q_t^h(\phi, a) = \mathbb{E}_{\xi \sim \pi_t}[r_t(\xi) | \phi_h = \phi, a_h = a]$.
  Define $A_t^h(\phi, a) = Q_t^h(\phi, a) - \mathbb{E}_{a' \sim \pi_t^h(\phi)}[Q_t^h(\phi, a')]$.
  for $\phi, a, h \in \Phi_h \times \mathcal{A} \times [H]$ **do**
    `// use no-regret algo for update`
    $\pi_{t+1}^h(a|\phi) \propto \pi_t^h(a|\phi) \exp\left(\eta A_t^h(\phi, a)\right)$.
  end for
end for
Return $\bar{\pi}$, uniform mixture of $\pi_{1:T}$.

Algorithm 4 SPO (Practical Version with Contexts)

Input: Iterations T, Preference fn. $\mathcal{P}$, Num. samples $k \geq 2$, Reinforcement learning algo. $\mathsf{RL} : \Pi \times \mathcal{D} \to \Pi$.
Output: Trained policy π.
Initialize $\pi_1 \in \Pi$.
for t in $1 \dots T$ **do**
  Observe context $x_t \sim \rho$ and sample $\xi_{1:k} \sim \pi_t(x_t)$.
  Compute $r_t(\xi_i) = \frac{1}{k-1} \sum_{j \neq i}^k \mathcal{P}(x_t, \xi_i, \xi_j)$.
  Set $r_i^h = r_t(\xi_i)/H, \forall i, h \in [k] \times [H]$.
  `// use PPO, TRPO, SAC ...`
  $\pi_{t+1} \leftarrow \mathsf{RL}(\pi_t, \mathcal{D} = \{(s_i^h, a_i^h, r_i^h)\}_{i \in [k]}^{h \in [H]})$.
end for
Return best of $\pi_{1:T}$ on validation data.

D. Compounding Errors in RLHF

A common concern in sequential prediction tasks is *compounding errors* (Ross et al., 2011). Consider, for example, trying to train a policy to drive laps around a track purely based on recorded demonstrations from an expert driver who always stays close to the center of their lane. For example, one could regress from recorded states to recorded actions, an approach known as behavioral cloning (BC, Pomerleau (1988)). However, if at test time, the agent makes a mistake and goes off the center of their lane, they might end up in a state they hadn't seen in their training data, have no idea what to do in this novel situation, make another error, and quickly spiral out of control. At a fundamental level, the agent's poor test-time performance is caused by the *covariate shift* in terms of state distribution between the offline training data and their own *induced* state distribution – a low training error does not necessarily correspond to a low test error. Thus, the standard solution is to allow the learner to actually try out actions in the environment, see where they end up, and learn to correct their mistakes. This approach is known as inverse reinforcement learning (IRL, Ziebart (2010)) and is known to prevent compounding errors (Swamy et al., 2021).

A natural question might be if the learner gets preferences rather than demonstrations as feedback, whether the same concerns arise. We now provide a simple, informal example of this.

Example D.1. *Consider the $H = 2$ problem of completing sentences of the form "*THE (A) ORBITS AROUND THE (B)*." where either blank can be filled in with utterances $\mathcal{A}$ = {*EARTH, SUN, MOON*}. We observe preferences of the form [*MOON, EARTH*] $\succ$ [*MOON, SUN*]. At $h = 1$, we are forced to play a policy π_ϵ that outputs* MOON *w.p.* $1 - \epsilon$ *and* EARTH *w.p.* ϵ*. At $h = 2$, we are allowed to choose between two policies: π_1 that always outputs* EARTH *and π_2 that outputs* EARTH *if the preceding word was* MOON *and* SUN *if the preceding word was* EARTH*. Observe that both $[\pi_\epsilon, \pi_1]$ and $[\pi_\epsilon, \pi_2]$ have the same probability of generating the preferred and dis-preferred generations and therefore will have the same value under **any** loss function that depends on its inputs purely via their off-policy likelihoods.*

We can think of π_ϵ as representing error from finite samples, imperfect optimization, or a limited function class. While hopefully any post-Copernican preference (or reward) model would be able to tell the difference between these two policies, offline approaches like DPO that simply compute likelihoods on off-policy data are unable to do so. [11] We emphasize that this is a fundamental issue with *all* offline approaches, rather than with a particular offline algorithm (Swamy et al., 2021). We conclude with a note that this problem only gets worse with longer task horizons as there are more timesteps to deviate. This is perhaps one of the reasons that offline approaches can sometimes under-perform interactive techniques (Zhu et al., 2023; Chen et al., 2024) and why, as of the writing of this paper, the world's most performant models are trained using interactive techniques (Team et al., 2023; OpenAI et al., 2023).

[11] An iterative application of DPO with batches of preferences collected frequently would likely mitigate this issue.

E. Additional Results

We now present additional results we did not have space for in the main paper:

- Figure 8 contains results on a larger set of control environments with preferences based on reward maximization.
- Figure 9 contains results on a larger set of control environments with noisy preferences based on reward maximization.
- Figure 10 contains results on a larger set of control environments with non-Markovian preferences.
- Figure 11 contains results for SPO on a larger set of bandit environments with intransitive preferences. Figure 12 contains the results for RM in the same setting.
- Figure 13 shows additional seeds for SPO the Ant environment with intransitive preferences.
- Figure 14 ablates the effect of different snippet lengths on RM sample efficiency.
- Figure 15 contains the effect of freezing the reward model during training on RM performance.

Number of queries to the preference oracle. We report the performance of an agent as a function of the total steps in the environment steps. We did not attempt to optimize performance of SPO or RM as a function of the number of calls to the preference oracle it makes. That said, in our experiments, SPO makes fewer oracle calls than RM. The total number of calls to an oracle made by SPO are the number of episodes collected times the queue length B, since each generated episode is compared against the previous B ones. In comparison, the total number of calls made by RM are the number of updates to the reward model times the batch size. For example, for the experiment with maximum-reward preferences (Figure 8), this is a total of **1M calls for SPO** and **2.5M calls for RM**.

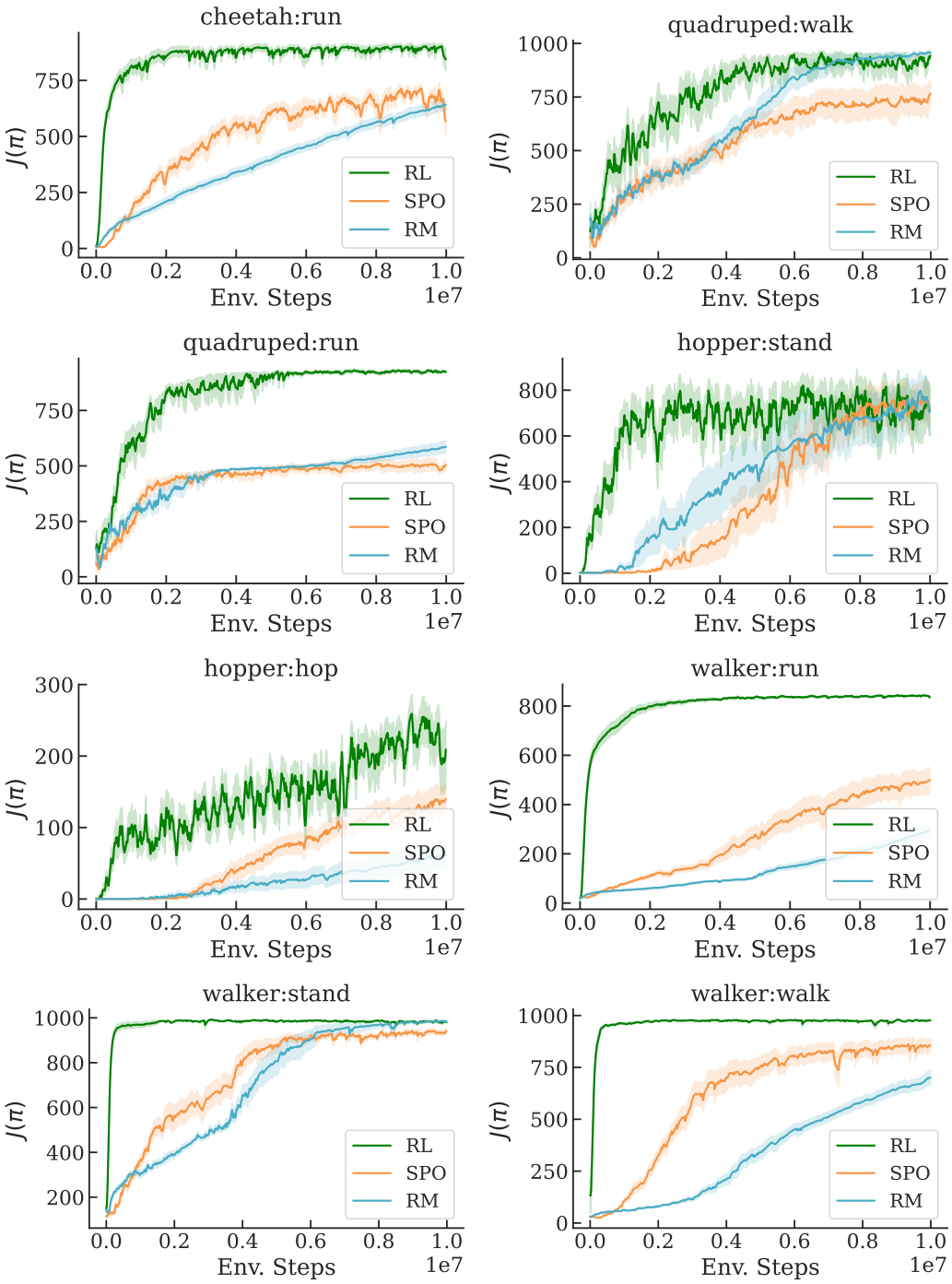


Figure 8: All results for maximum reward preference experiments.

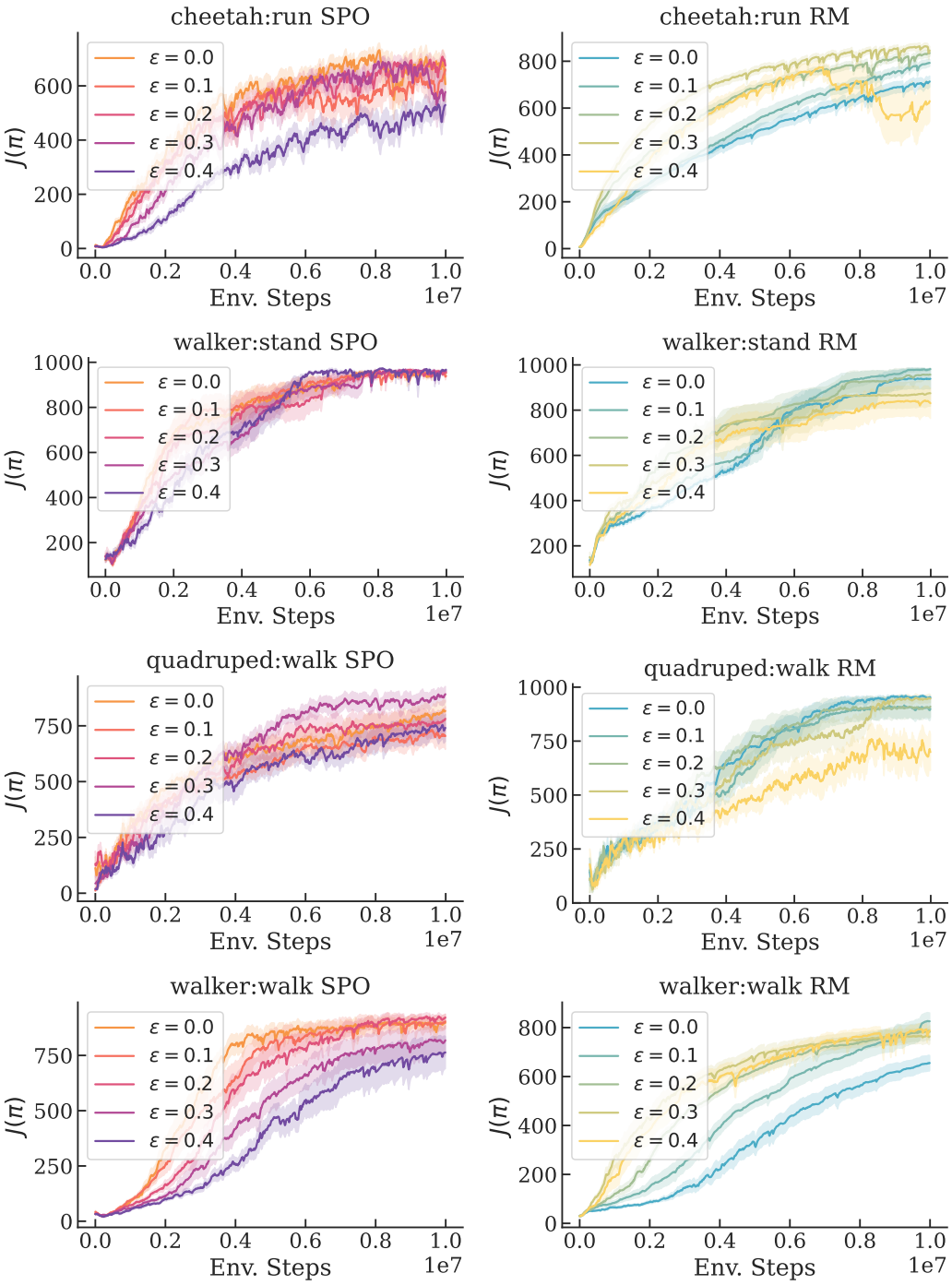


Figure 9: All results for noisy preference experiments.

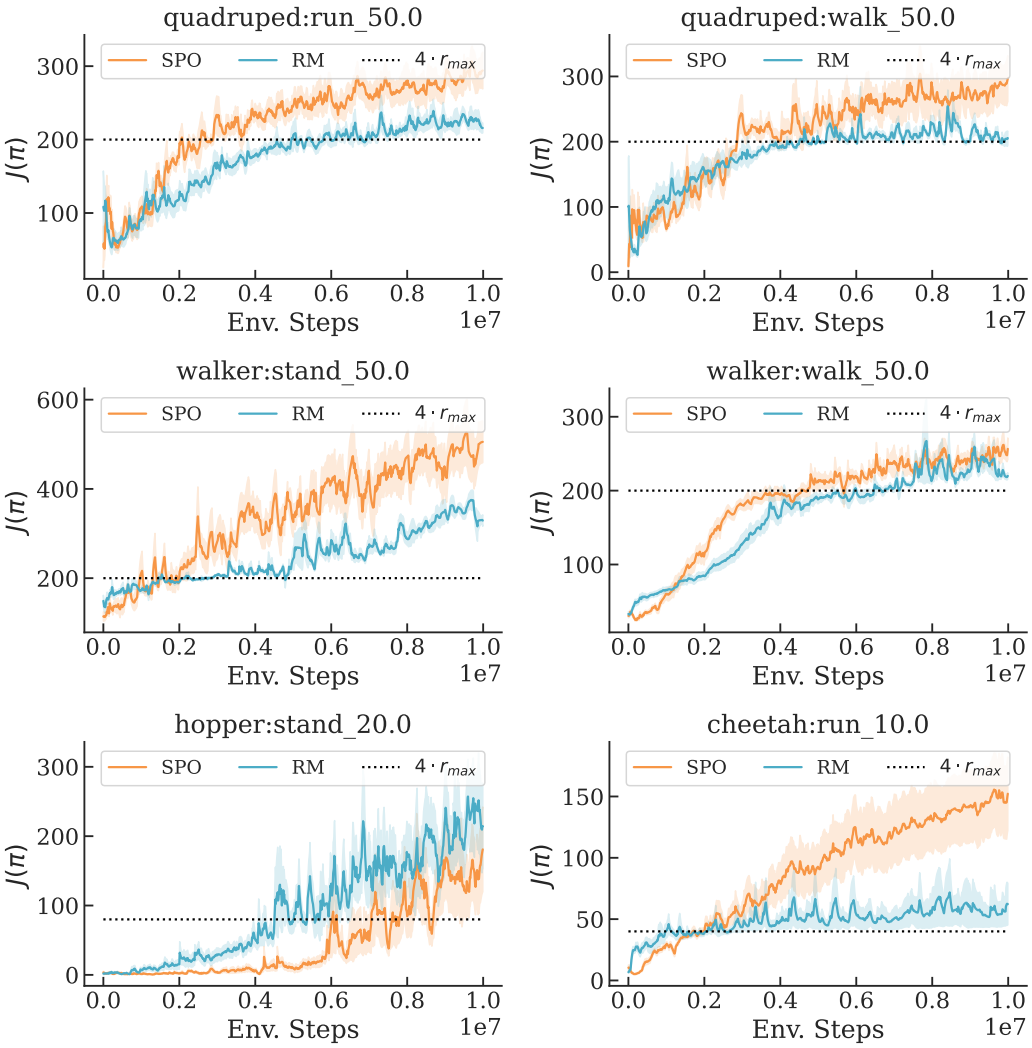


Figure 10: All results for non-markovian preference experiments.

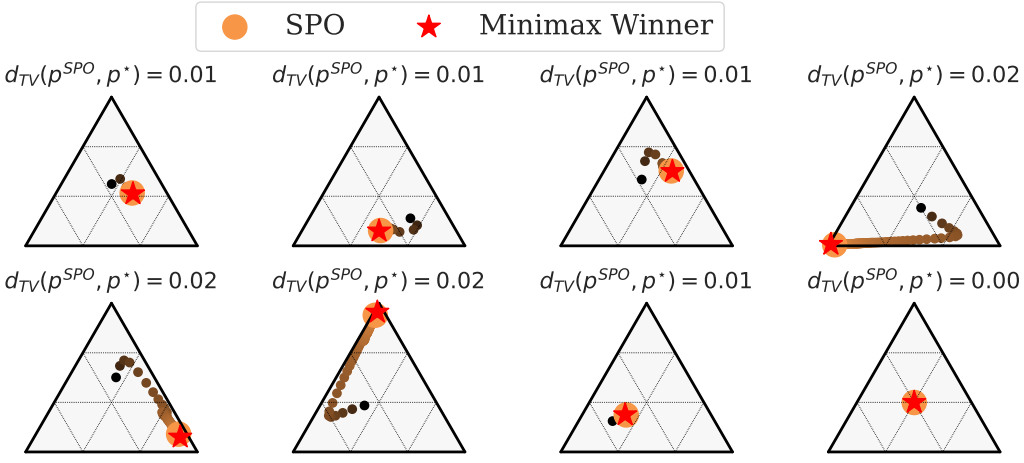


Figure 11: Convergence of SPO on bandit instances with non-transitive preferences.

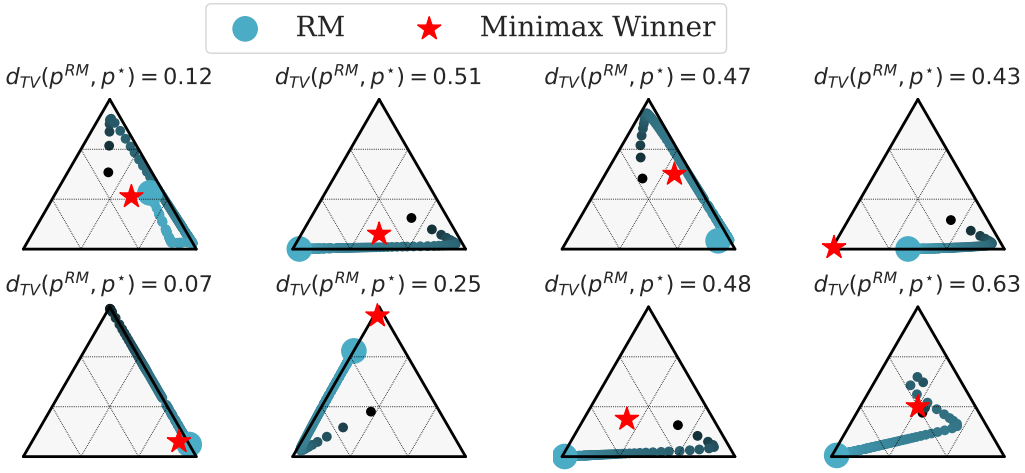
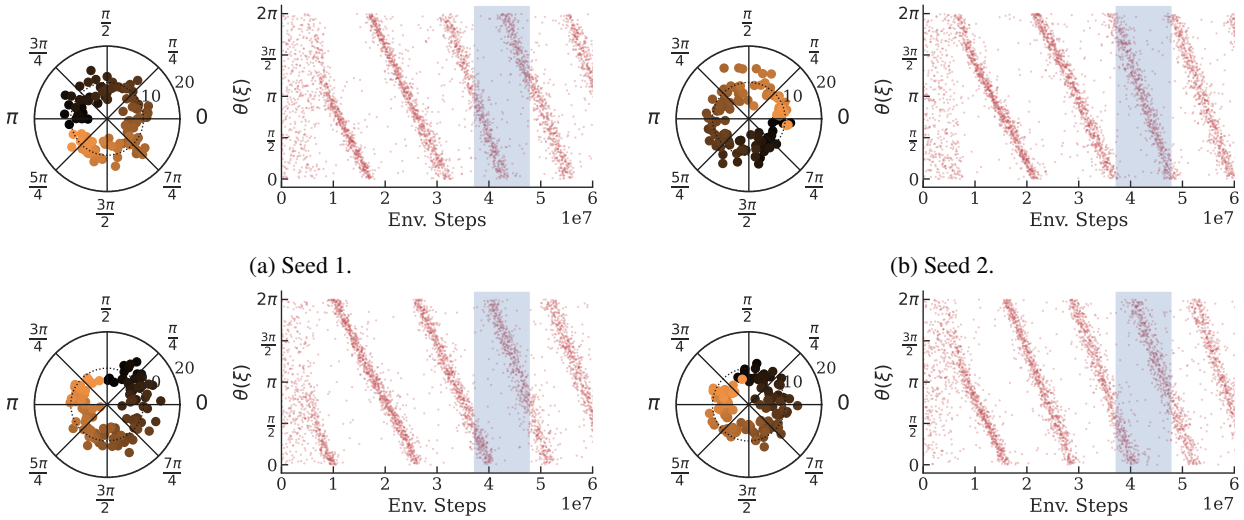


Figure 12: Convergence of RM on bandit instances with non-transitive preferences.



(a) Seed 1.

(b) Seed 2.

(c) Seed 3.

(d) Seed 4.

Figure 13: Additional results for intransitive preference experiments with Gym Ant-v3 environment. Here, we present results with 4 seeds with these trends holding across all the 10 seeds we experimented with. As presented in the main paper, in the polar plots, each dot corresponds to 100k timesteps of training (where the darker shade represents earlier environment steps and lighter shade represent later environment steps). We see our agent traces out a circle of radius 10 (qualitatively the behavior we expect from MW) on average. Alongside the polar plots, we plot the agent's angle through the first 60 million steps of training with the shaded region representing the cross section of iterations where the agent's position and angle are plotted in the corresponding polar plot.

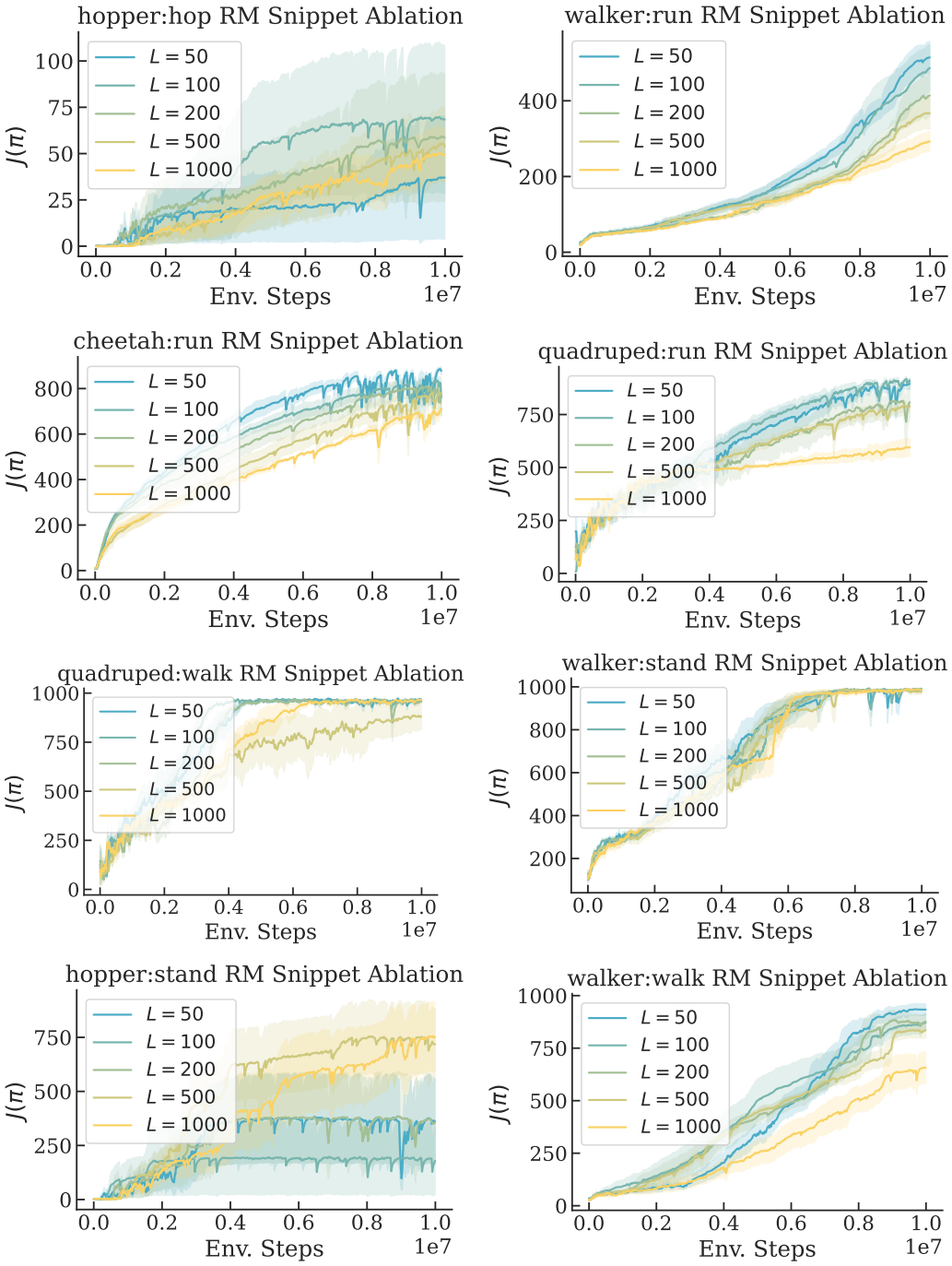


Figure 14: All results for reward model snippet length ablations.

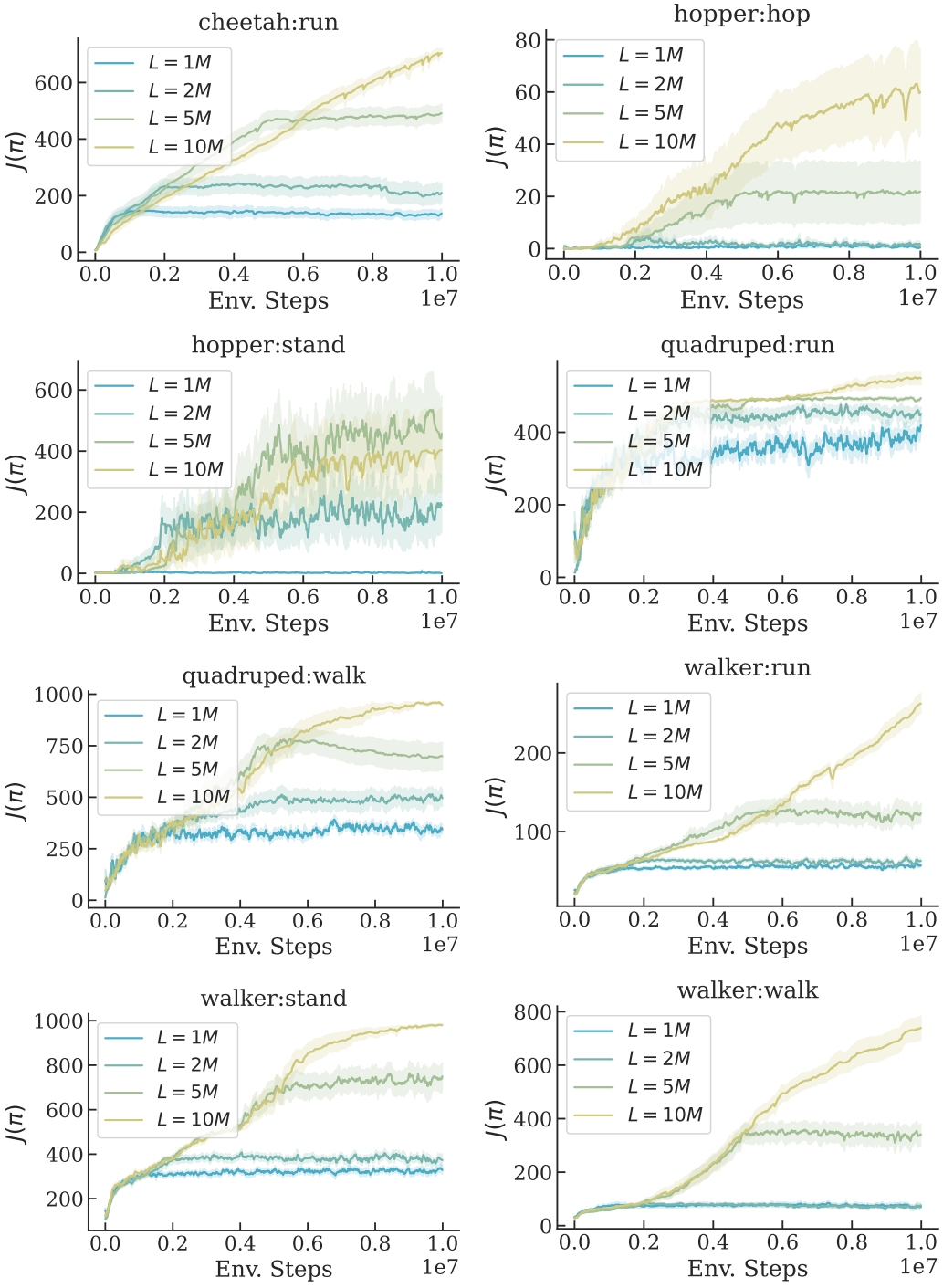


Figure 15: All results for reward model freezing experiments.

F. Experimental Details

Contents

F.1. Discrete action experiment in Figure 5 (a)

Problem Setup. We use a 3-armed bandit with different preference functions that each induce a different (unique) Minimax Winner. All our preference functions have the following form, represented as an $|\mathcal{A}|$ by $|\mathcal{A}|$ matrix

$$\mathcal{P} = \begin{bmatrix} 0 & c & -b \\ -c & 0 & a \\ b & -a & 0 \end{bmatrix}$$

where $a, b, c > 0$ are parameters. The Minimax Winner for this preference function is $p^\star \propto \begin{bmatrix} a \\ b \\ c \end{bmatrix}$. This can be verified easily since $p^\star \mathcal{P} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ and thus any opponent strategy yields the same value 0. For $a = b = c$, this is the preference in the popular Rock-Paper-Scissors game. More generally such preference functions can commonly occur when we estimate the preference from a population of users that each just have a preference between two of the three actions. Assume we have 3 sub-populations that each make up a fraction a, b and c of the total population, respectively. Then the average preference in the total population corresponds to $\mathcal{P}$

$$a \times \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} + b \times \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} + c \times \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \mathcal{P},$$

where each of the 3 matrices corresponds to the preference in the respective subpopulation.

Common Parameters. We use the PPO implementation in Hoffman et al. (2020) with learning rate 10^{-4}. For the policy network, we use 2 hidden layer with 128 nodes each and ReLU activation. Since the purpose of this experiment is to verify whether a method can approach the Minimax Winner, we run each algorithm for a very large number of of steps, 50M and report the average action choice during the entire learning procedure.

SPO Parameters. We use a queue length of $B = 1000$ trajectories for computing the average preference of each trajectory. We further found that the default entropy cost of 10^{-4} for PPO in ACME (Hoffman et al., 2020) to work well.

Reward Model Parameters. We use the same architecture for the reward model as for the policy network with a one-hot encoding of the action. We update the reward model every 16 policy updates, use a batch size of 64, and a learning rate of $3 \cdot 10^{-4}$. We further use an increased entropy cost of 10^{-2} for PPO. We found that less frequent reward-model updates and a smaller entropy cost would lead to premature convergence at arbitrary deterministic policies. Since we want to learn non-trivial MW strategies and theory suggests that a policy-dependent reward model could achieve that (see Equation 7 and discussion around it), We also experimented with more frequent reward-model updates. These led to instabilities in the learning process, however.

F.2. Continuous Control Experiments

We use the SAC implementation in Hoffman et al. (2020) for all of our continuous control experiments. We use the same SAC hyperparameters for all methods, other than the fact that we use 3e-4 rather than 3e-5 as the learning rate for vanilla SAC. We use Adam for all optimization. We use three layer networks of width 256 for all function approximation. We use ReLU activations for the actor and critic.

PARAMETER	VALUE
MAX REPLAY SIZE	1000000
MIN REPLAY SIZE	10000
BATCH SIZE	256
γ	0.99
τ	0.005
N STEP	1
POLICY LEARNING RATE	3E-5
BATCH SIZE	256
SAMPLES PER INSERT	256
SAMPLES PER INSERT TOLERANCE RATIO	0.1
NUM SGD STEPS PER STEP	1

Table 2: Hyperparameters for SAC.

SPO Parameters. We use a queue length of $B = 10$ and a maximum replay buffer size of $1M$ for non-Markov and intransitive preferences. For max-reward preferences (excl. and incl. noise), we use a queue length of $B = 100$ due to the higher noise in the preferences. We further use a shorter maximum replay buffer size of $100k$ to avoid preference information becoming too stale due to the increased queue length.

Reward Model Parameters. We use Leaky ReLU activations with a final tanh (following Lee et al. (2021a)) for the reward model. We update the reward model every 256 policy updates, use a batch size of 64, and a learning rate of 3e-4.

F.3. Continuous Control Experiments with Intransitive Preferences

We use the MuJoCo Gym (Brockman et al., 2016) Ant-v3 environment as the base environment. The hyper-parameters for SAC are identical to what is described above except for having a fixed entropy coefficient of 1e-4 since the learned policy must exhibit stochasticity to capture the MW; we use a queue size of 100. The preference function we design is composed of a distance and angular preference. The angular preference makes the trajectory lose to an angle of $\theta = \pi/4$ in front of them. The distance component encourages the agent to traverse non-trivial distance from the origin until it hits a certain threshold. We found the distance component to be necessary in order for the agent to maintain effective control of the angle as it is very easy to shift angles when the agent is very close to the origin. The Python code for the preference function is written below.

```
def intransitive_ant_preference(traj_1, traj_2):
    return 0.3 * distance_preference(traj_1.radius, traj_2.radius, 10.0) +
           0.7 * angular_preference(traj_1.angle, traj_2.angle, math.pi/4)

def angular_preference(traj_1_angle, traj_2_angle, angle):
    difference = math.fmod(traj_1_angle + angle/2.0 - traj_2_angle, 2 * math.pi)
    return difference < theta/2.0 or difference > 2 * math.pi - theta/2.0

def distance_preference(traj_1_dist, traj_2_dist, dist_threshold):
    if traj_1_dist > dist_threshold and traj_2_dist > dist_threshold:
        return 1.0
    else:
        return 1.0 if traj_1_dist > traj_2_dist else 0.0
```