Active Preference Optimization for Sample Efficient RLHF

Nirjhar Das[1] (✉), Souradip Chakraborty[2], Aldo Pacchiano[3], and Sayak Ray Chowdhury[4]

[1] Indian Institute of Science, Bangalore, India `nirjhardas@iisc.ac.in`
[2] University of Maryland, College Park, USA `schakra3@umd.edu`
[3] Boston University, USA `pacchian@bu.edu`
[4] Indian Institute of Technology Kanpur, India `sayakrc@cse.iitk.ac.in`

Abstract. Large Language Models (LLMs) aligned using Reinforcement Learning from Human Feedback (RLHF) have shown remarkable generation abilities in numerous tasks. However, collecting high-quality human preferences creates costly bottlenecks in practical deployments, and hence, training data are often budgeted. In these scenarios, it is crucial to collect training data (e.g., contexts, a pair of generations for each context, and a preference indicating which generation is better) carefully, yet most of the existing methods sample contexts uniformly at random from a given collection. Given this, under the Bradley-Terry-Luce preference model and with a small budget of training data, we show that uniform sampling of contexts could lead to a policy (i.e., an aligned model) that suffers a constant sub-optimality gap from the optimal policy. This highlights the need for an adaptive context sampling strategy for effective alignment under a small sample budget. To address this, we reformulate RLHF within the contextual preference bandit framework, treating generations as actions, and give a nearly complete characterization of the sub-optimality gap in terms of both lower and upper bounds. First, when the action set is a d-dimensional hypercube and the number of samples is T, we show an $\Omega(d/\sqrt{T})$ lower bound. Next, we propose an algorithm, *Active Preference Optimization* (`APO`), that iteratively collects preferences for the most uncertain contexts. We show that the sub-optimality gap of the policy learned via `APO` matches the lower bound up to a log factor and a non-linearity constant. Finally, we perform experiments on practical datasets to validate `APO`'s efficacy over existing methods, establishing it as a sample-efficient and cost-effective solution for LLM alignment.

1 Introduction

Reinforcement Learning from Human Feedback (RLHF) has proven highly effective in aligning Large Language Models (LLMs) with human preferences [6,20,8]. This approach involves collecting extensive data, each comprising a context (e.g., a movie review), a pair of generations (e.g., completions of the review), and a preference indicating which generation is better than the other. First, a reward model is trained to classify preferred generations, and subsequently, a language model policy is trained using RL (e.g., Proximal Policy Optimization [28]) to output high-reward generations while minimizing divergence from a reference policy. Given the practical success, recent theoretical advances have been made in training reward models as well as aligning policies from pairwise comparisons [26,5,32]. In these settings, the learner doesn't have any control over selecting the contexts, and the aim is to minimize cumulative loss or regret due to not knowing the ground-truth reward or the optimal policy in advance. However, in the case of aligning LLMs using RLHF, the learner has control over both the contexts and actions, i.e., the contexts and generations for which preference data needs to be collected, yet most of the existing RLHF algorithms pick contexts uniformly at random from a given pool [29,20]. This is followed by first generating a pair of responses for each sampled prompt based on a supervised fine-tuned (SFT) policy and then sending all the pairs to a human labeler to collect preferences.

The success of RLHF hinges on the quality of human preferences. This could create costly bottlenecks in practical deployments since high-quality preferences are expensive to collect as this demands a certain level of expertise from labelers. Hence, there is often a budget on the number of contexts and associated generation pairs that could be sent to expert labelers for comparison. While uniform sampling of contexts as a simple approach has been proven effective for aligning LLMs so far, one is bound to ask whether this is a good enough strategy, especially given a fixed budget for labeling. Or do we need potentially more involved sampling strategies to deliver better model alignment under budget constraints? Such algorithms need to be sample efficient as high-quality samples are expensive to obtain, while they should not compromise on the performance of the aligned policy. This work takes a step in developing theory and algorithms for RLHF under a small sample budget.

1.1 Overview of Main Results

We first show that the naive way of collecting preferences by choosing contexts uniformly at random can lead to wastage of samples under the Bradley-Terry-Luce (BTL) preference model characterized by a finite dimensional parameter [3,15].

Result 1 (Constant sub-optimality under uniform context sampling) *There exists an instance of the alignment problem for which an algorithm that (i) collects preferences by sampling contexts uniformly at random, (ii) learns a reward model by maximizing likelihood of the preferred generations, and (iii) trains a greedy policy w.r.t. the learnt reward model suffers an $\Omega(1)$ sub-optimality gap*[5] *with high probability when the sample budget is smaller than number of contexts.*

To the best of our knowledge, this is the first provable negative result for the alignment performance of RLHF algorithms that sample contexts uniformly under a small sample budget. This necessitates designing of RLHF algorithms that adaptively sample contexts, with an aim to improve alignment of the learned policy

[5] Measured by the maximum difference between latent rewards of the optimal policy and the trained policy over the context set.

with human preferences. Our next result benchmarks the alignment performance of any RLHF algorithm under the BTL preference model when there is no restriction on how contexts can be sampled.

Result 2 (Lower bound on sub-optimality for any sampling strategy) *For any RLHF algorithm there exists an instance of the alignment problem for which the policy that the algorithm outputs after collecting T samples (contexts, generations and preferences) would suffer a sub-optimality gap $\Omega\big(d/\sqrt{T}\big)$, where d is the dimension of the parameter characterizing the BTL model.*

This is the first theoretical lower bound on the alignment performance of RLHF algorithms, which has been crucially missing in the literature. This result effectively eliminates the possibility of achieving better than sub-linear convergence under a finite sample budget. Next, we propose an algorithm – *Active Preference Optimization* (`APO`) – that achieves this sub-optimality gap by iteratively sampling the most uncertain contexts and collecting preferences for their generation pairs.

Result 3 (Upper bound on sub-optimality for `APO`) *The sub-optimality gap of the policy learned via `APO` after collecting T samples scales as $\tilde{O}\big(d\sqrt{\kappa/T}\big)$ with high probability, where d is the parameter dimension and κ is a problem-dependent nonlinearity constant.*

Next, we generalize our result from parameterized BTL model to non-parametric preference models with function approximation. We propose an analogue of `APO` albeit with non-trivial modifications, namely `APO-Gen`, that achieves a similar sub-optimality gap (see Subsection 4.2 and Appendix D). This is the first known upper bound on the alignment performance of an active context selection strategy under generic preference models, which recovers the result for the parameterized BTL model as a special case.

Result 4 (Upper bound on sub-optimality under general preferences) *The sub-optimality gap of the policy learned via `APO-Gen` after collecting T samples scales as $\tilde{O}\big(\sqrt{d_{\mathcal{E}}\log(\mathcal{N}T)/T}\big)$ with high probability, where $\mathcal{N}$ and $d_{\mathcal{E}}$ measure the complexity of the underlying preference model.*

Empirical evidence. For practical purposes, we propose a batch version of `APO` to make it computationally more efficient (see Section 5). We experiment with GPT-2 on IMDb sentiment dataset [16] and demonstrate significant improvement in LLM alignment over uniform context sampling and prior baselines [18,19]. We show similar improvement in the performance of the aligned policy on Anthropic-HH dataset [2] with Gemma-2b. Our work contributes towards a sample-efficient and practical solution to preference data collection for RLHF.

1.2 Comparison with Prior Work

Active learning in the context of Preference-based Reinforcement Learning (PbRL) [26,5], which is used as a theoretical framework for RLHF, has received some attention recently. In PbRL literature, the problem of learning the reward function by actively querying the human labeller has been considered in [24,4,14,10,31]. The work [14] provides variance and entropy-based heuristics for learning the optimal policy without providing any provable guarantee. On the other hand, in [24], the authors design an algorithm for learning the reward function by actively synthesizing trajectories via a volume removal scheme over the distribution of the unknown parameter. They show that under certain strong assumptions, their algorithm makes progress in reducing the uncertainty over the parameter distribution. Similarly, [4] also aims at actively learning the reward function using model epistemic uncertainty as well as the entropy estimate of acquiring a data point but does not provide any provable guarantee. Hence, both of these are orthogonal to our work since they consider learning the reward function, whereas we focus on learning a policy - while learning a good reward function is sufficient for learning a good policy, it is not at all necessary.

In [31], the authors propose a pure exploration strategy for the PbRL problem that finds an ε-optimal policy with $O(\kappa^2 d^2/\varepsilon^2)$ queries to the human labeler.

Instead of modeling RLHF as a PbRL problem, we model it as a contextual dueling bandit problem, where contexts model prompts and actions model generations of an LLM. This necessitates a strategy for not only actively selecting actions but also selecting contexts actively. This is a major point of departure from most active-learning based PbRL works and from pure exploration in dueling bandits literature [7,17]. It is unclear apriori what should be the optimal strategy to select the context, towards which we show that a design-matrix-based exploration bonus is sufficient (see Section 4). Moreover, a direct comparison shows that our result (Theorem 3) is tighter in terms of κ. In [10], the authors consider an active learning-based approach to regret minimization in the contextual dueling bandits. Again, [10] does not address the question of context selection but rather allows contexts to be adversarially presented to the learner, who only chooses action given the context and whether to observe the labeler feedback.

Existing works closest to ours are [19,18], which also investigate the problem of actively selecting prompts and generations for RLHF in LLMs. [19] proposes an algorithm that actively selects contexts using a heuristic based on generation uncertainty, but they do not give any theoretical guarantee for the proposed method. [18] proves a sub-optimality gap bound for an active context selection strategy that goes down sub-linearly with the number of samples. However, they assume a strong restrictive condition on the preference model, which *doesn't hold in general* for the BTL model. We remove this restrictive assumption and provide an improved guarantee (see Remark 2 for a detailed comparison).

2 Problem Setup

We have a set of contexts $\mathcal{X}$ and a set of possible actions per context $\mathcal{A}$. To learn using preference feedback, the agent selects $x \in \mathcal{X}$ and $a, a' \in \mathcal{A}$ to present to a human labeller, who then reveals a binary preference y that takes value 1 if a wins over a' and 0 otherwise. We assume that given (x, a, a'), y is sampled from the Bradley-Terry-Luce (BTL) preference model [3,15] with r^* as the latent (unknown) reward function, i.e.,

$$\mathbb{P}\left[y=1|x,a,a';r^*\right] = \frac{\exp(r^*(x,a))}{\exp(r^*(x,a)) + \exp(r^*(x,a'))} \ ,$$

The goal of the agent is to first learn r^* over T rounds of sequential interaction with the labeller, collecting dataset $\mathcal{D} = (x_s, a_s, a'_s, y_s)_{s=1}^T$, and then employ the learned reward to train a policy $\pi : \mathcal{X} \to \mathcal{A}$, which will eventually fetch high latent rewards $r^*(x, \pi(x))$.

In this work, we consider linear latent rewards $r^*(x,a) = \phi(x,a)^\top \theta^*$, where $\theta^* \in \mathbb{R}^d$ is the unknown reward parameter, and $\phi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$ is some known and fixed feature map. For instance, such a ϕ can be constructed by removing the last layer of a pre-trained language model, and in that case, θ^* corresponds to the weights of the last layer. With this model, for any $\theta \in \mathbb{R}^d$, one can equivalently write the probability of sampling $y_s = 1$ given (x_s, a_s, a'_s) as

$$\mathbb{P}\left[y_s=1|x_s,a_s,a'_s;\theta\right] = \sigma\big((\phi(x_s,a_s) - \phi(x_s,a'_s))^\top\theta\big) = \sigma(z_s^\top \theta) \ ,$$

where $\sigma(w) = \frac{1}{1+e^{-w}}$ is the sigmoid function and $z_s = \phi(x_s,a_s) - \phi(x_s,a'_s)$ is the feature differenc of actions a_s and a'_s for context x_s.

With this, the latent reward parameter θ^* is typically estimated by minimizing the binary cross entropy loss (log-loss) [20], which is equivalent to *maximum likelihood estimation* (MLE). Specifically, At round t, the MLE of θ^* is computed as $\widehat{\theta}_t = \operatorname{argmin}_{\theta\in\Theta} \mathcal{L}_t(\theta)$ using the preference dataset $\{(x_s, a_s, a'_s, y_s)\}_{s=1}^{t-1}$, where log-loss $\mathcal{L}_t(\theta)$ is given by

$$\mathcal{L}_t(\theta) = -\sum\nolimits_{s=1}^{t-1} y_s \log(\sigma(z_s^\top\theta)) + (1-y_s)\log(1-\sigma(z_s^\top\theta)). \qquad (1)$$

The above optimization problem is convex if we let the constraint set $\Theta \subset \mathbb{R}^d$ to be convex, and hence can be solved using standard algorithms [9].

Performance Measure. Our goal is to learn a policy over the collected data $\mathcal{D}$, which has high rewards or, equivalently, low sub-optimality. Formally, the sub-optimality gap of a policy π_T trained on the dataset $\mathcal{D}$ is defined as

$$R(T) = \max\nolimits_{x\in\mathcal{X}} \max\nolimits_{a\in\mathcal{A}} \left\{r^*(x,a) - r^*(x,\pi_T(x))\right\} . \qquad (2)$$

Here, our policy π_T competes with the *Condorcet* winner for a given context – an action that fetches a higher reward than all other actions. The sub-optimality gap is the worst possible difference in rewards over the set of contexts. Prior work [18] competes against the *Borda* winner – an action that fetches a higher reward on average than another randomly chosen action – a weaker competitor (the *Condorcet* winner is also the *Borda* winner but not the other way around).

Remark 1. To the best of our knowledge, common practical implementations of the RLHF pipeline use the following method: (i) remove the top layer of the LLM and convert it into an encoder, (ii) append a new linear layer on top of it, and (iii) output the logit score. Hence, the linear reward assumption is not restrictive in the sense that we only train the linear layer, keeping the encoder fixed. In practice, however, one needs to train the encoder, and hence, a more general function class needs to be considered. To this end, we generalize this setup to preference models with bounded class complexities, removing the need for explicit linear reward models (see Section 4 and Appendix D for details).

3 Lower Bounds

We first illustrate the pitfall of a learner who samples contexts uniformly. We characterize such a learner in preference-based learning/RLHF and then show that such a learner can suffer a constant sub-optimality gap under budget constraints.

Definition 1 (Uniform Learner). *Say an algorithm* `Alg` *samples T contexts uniformly at random from a set $\mathcal{X}$ and for each context x_t, picks two actions a_t, a'_t from a set $\mathcal{A}$. For each chosen triplet (x_t, a_t, a'_t),* `Alg` *queries the preference model parameterized by θ^* and observes a stochastic preference $y_t \in \{0,1\}$ between the actions.* `Alg` *then solves an MLE on this data to obtain $\widehat{\theta}$, and learns a greedy policy with respect to $\widehat{\theta}$. We call such an algorithm* `Alg` *a Uniform Learner.*

Theorem 1 (Lower bound for uniform context sampling). *There exists a problem instance $(\mathcal{X}, \mathcal{A}, \theta^*)$ for which the policy learnt by a Uniform Learner* `Alg` *under the budget $T \ll |\mathcal{X}|$ suffers $\Omega(1)$ sub-optimality gap with high probability.*

Proof (Sketch). We show the result for $d = 2$ for simplicity. The main idea is to divide the set of contexts into two groups—*good* and *bad*. Further, every context has only two actions, a and a'. The *good* group has a large number of contexts, so the uniform learner mostly samples contexts from this group. For all context x in the *good* group, the feature difference $\phi(x,a) - \phi(x,a') = z_g$ is the same. For *bad* contexts, this feature difference is z_b, such that $\langle z_b, z_g \rangle < 0$. Finally, θ^* is taken as the angle-bisector of z_g and z_b.

From this construction (Fig. 2 in Appendix A), we have that a gets a higher reward for every context. Then, we show that the uniform learner only samples context-actions corresponding to z_g when the sample budget T is much smaller than the number of contexts. Under this scenario, the MLE estimate of the uniform learner correctly classifies the reward of a to be higher than that of a' for all the *good* contexts, but it wrongly classifies the rewards for the *bad* contexts. Thus, for *bad* contexts, the uniform learner suffers a constant suboptimality gap. Details are in Appendix A. □

Theorem 1 effectively shows that the uniform learner cannot make efficient use of the sampling budget because its performance may not increase with an increasing budget. Now, it is essential to characterize the limits of learning in this setting. To this end, we prove a lower bound on the sub-optimality gap of any algorithm with no restriction on how contexts and action pairs can be sampled. Note that standard regret lower bounds for dueling [25] and logistic bandits [1] are not applicable here because we bound the sub-optimality gap and not regret.

Theorem 2 (Lower Bound for any sampling strategy). *Let $\mathcal{X}$ be a finite set of contexts, $\Theta = \{-\frac{1}{\sqrt{T}}, \frac{1}{\sqrt{T}}\}^d$, $\mathcal{A} = \{-\frac{1}{2}, \frac{1}{2}\}^d$. Then, for any algorithm, there exists a parameter $\theta^* \in \Theta$ such that sub-optimality gap of a policy learnt by the algorithm after collecting T samples satisfies*[6]

$$\mathbb{E}_{\theta^*}\left[R(T)\right] \geq \Omega\left(d/\sqrt{T}\right) .$$

Proof (Sketch). Without loss of generality, choose any $x \in \mathcal{X}$. Let $\pi_T(x)$ denote the action chosen by the policy learnt using T samples. Define the event $\mathcal{E}_{\theta,i} = \{\text{sign}(\pi_{T,i}(x)) \neq \text{sign}(\theta_i)\}$, for all $i \in [d]$, where $\pi_{T,i}(x)$ and θ_i are the i-th coordinates of $\pi_T(x)$ and the parameter $\theta \in \Theta$, respectively. Note that under the event $\mathcal{E}_{\theta,i}$, the algorithm suffers a sub-optimality of $\frac{1}{\sqrt{T}}$ for the i-th coordinate. We need to lower bound the probability of this event. To this end, let $\theta' \in \Theta$ be such that $\theta'_i = -\theta_i$ and $\theta'_j = \theta_j$ for $j \neq i$. Note that $\mathcal{E}^c_{\theta,i} = \mathcal{E}_{\theta',i}$. Therefore, from Lemma 3 (Appendix B), we have, $\mathbb{P}_\theta[\mathcal{E}_{\theta,i}] + \mathbb{P}_{\theta'}[\mathcal{E}_{\theta',i}] \geq \frac{1}{2}\exp(-D_{KL}(\mathbb{P}_\theta, \mathbb{P}_{\theta'}))$.

Next, we need an upper bound on $D_{KL}(\mathbb{P}_\theta, \mathbb{P}_{\theta'})$. Using [12, Lemma 15.1] and Taylor expansion of log-sigmoid, we obtain $D_{KL}(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) \leq \frac{1}{8}\mathbb{E}_\theta\left[\sum_{t=1}^T \langle z_t, \theta - \theta'\rangle^2\right]$, where $z_t = a_t - a'_t$. Since $z_t \in [-1,1]^d$, and θ, θ' are equal in every coordinate except the i-th one, we have, $D_{KL}(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) \leq \frac{1}{8}\sum_{t=1}^T \mathbb{E}_\theta[(2z_{t,i}\theta_i)^2] \leq \frac{1}{2}$. Hence,

$$\frac{1}{|\Theta|}\sum\nolimits_{\theta\in\Theta}\sum\nolimits_{i=1}^d \mathbb{P}_\theta[\mathcal{E}_{\theta,i}] \geq \frac{d}{4}\exp(-\frac{1}{2}) .$$

Therefore, there exists a $\theta^* \in \Theta$ such that $\sum_{i=1}^d \mathbb{P}_{\theta^*}[\mathcal{E}_{\theta^*,i}] \geq \frac{d}{4}\exp(-\frac{1}{2})$. Finally, it is easy to see that the expected sub-optimality gap is lower bounded by the expected gap for context x. Hence, we have the following chain of inequalities:

$$\mathbb{E}_{\theta^*}[R(T)] \geq \mathbb{E}_{\theta^*}\left[\sum\nolimits_{i=1}^d \mathbb{1}[\mathcal{E}_{\theta^*,i}]\cdot 2|\theta^*_i|\right] = \frac{2}{\sqrt{T}}\sum\nolimits_{i=1}^d \mathbb{P}_{\theta^*}[\mathcal{E}_{\theta^*,i}] \geq \frac{d\exp(-1/2)}{2\sqrt{T}}$$

which completes the proof. Details are in Appendix B. □

To the best of our knowledge, Theorem 2 gives the first lower bound for general active-learning algorithms, which was missing in prior work [18,19]. Now, the immediate question is whether one can design an algorithm that learns a policy whose sub-optimality gap matches this lower bound. In the next section, we present an algorithm *Active Preference Optimization* (`APO`) that achieves this up to log factors and an instance-dependent non-linear factor.

4 Our Approach: Active Preference Optimization

At each round t, `APO` (Algorithm 1) proceeds by computing the MLE estimate $\widehat{\theta}_t$ based on the data obtained in the past $t-1$ steps (1). Based on $\widehat{\theta}_t$, our goal

[6] Expectation is over the randomness of $(x_1, a_1, a'_1, y_1, \ldots, y_T)$ under hypothesis θ^*.

Algorithm 1 `APO`: Active Preference Optimization

```
Require: Context set $\mathcal{X}$, action set $\mathcal{A}$, feature map $\phi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$, regularization
    $\lambda > 0$, and failure probability $\delta \in (0, 1]$. Initialize $\widehat{\theta}_1 = 0$.
1: for $t = 1, \ldots, T$ do
2:     Choose the triplet $(x_t, a_t, a'_t)$ using (3) and (5).
3:     Observe preference feedback $y_t \sim \mathtt{Ber}\big(\sigma(z_t^\top \theta^*)\big)$, where $z_t = \phi(x_t, a_t) - \phi(x_t, a'_t)$.
4:     Compute reward estimate $\widehat{\theta}_{t+1}$ that minimizes the constrained log-loss (1).
5:     Compute (scaled) design matrix $H_{t+1}(\widehat{\theta}_{t+1})$ via (4).
6: Compute final policy $\pi_T(x)$ using (6).
```

is to maximize exploration. To do this, for a context $x \in \mathcal{X}$, we compute the uncertainty $b_t(x, a, a')$ for each action (a, a') available for that context and choose the one which maximizes this, i.e., we choose the pair

$$(a_t(x), a'_t(x)) = \mathrm{argmax}_{(a,a') \in \mathcal{A} \times \mathcal{A}} b_t(x, a, a'), \tag{3}$$

where $b_t(x, a, a') = \|\phi(x, a) - \phi(x, a')\|_{H_t^{-1}(\widehat{\theta}_t)}$ and $H_t(\widehat{\theta}_t)$ is a matrix that describes a confidence ellipsoid around the unknown reward parameter θ^* after $t - 1$ steps of data collection. For any $\theta \in \Theta$, this is defined as

$$H_t(\theta) = \nabla^2 \mathcal{L}_t(\theta) + \lambda \mathbf{I}_d = \sum_{s=1}^{t-1} \dot{\sigma}(z_s^\top \theta) z_s z_s^\top + \lambda \mathbf{I}_d \ , \tag{4}$$

where $z_s = \phi(x_s, a_s) - \phi(x_s, a'_s)$ is the feature difference for the triplet (x_s, a_s, a'_s). Intuitively, the confidence ellipsoid keeps shrinking along whichever direction (in $\mathbb{R}^d$) we decide to explore. Thus, for a given context x, choosing the pair $(a_t(x), a'_t(x))$ maximally reduces the uncertainty among all other possible action duels. However, our algorithm picks not only the action pair that maximizes uncertainty but also the context that increases it the most, i.e.,

$$x_t = \mathrm{argmax}_{x \in \mathcal{X}} \, b_t(x, a_t(x), a'_t(x)) \ . \tag{5}$$

This is a crucial step in our approach that ensures that the uncertainty of the reward function over all contexts decreases at a fast rate, which in turn ensures a low sub-optimality gap. After T rounds, we define $\theta_T = \frac{1}{T} \sum_{s=1}^{T} \widehat{\theta}_t$ as the average of all the past parameter estimates. Our final policy π_T for any context $x \in \mathcal{X}$ is to play the action that maximizes the reward parameterized by θ_T, i.e.,

$$\pi_T(x) = \mathrm{argmax}_{a \in \mathcal{A}(x)} \, \widehat{r}_T(x, a) = \mathrm{argmax}_{a \in \mathcal{A}(x)} \, \phi(x, a)^\top \theta_T \ . \tag{6}$$

4.1 Suboptimality Gap of `APO`

We make the following assumption, which is standard in the literature [32,23].

Assumption 1 (Boundedness) *(a) θ^* lies in the set $\Theta = \{\theta \in \mathbb{R}^d | \langle \mathbf{1}, \theta \rangle = 0, ||\theta|| \leq S\}$. (b) Features are bounded, i.e., $||\phi(x, a)|| \leq 1, \forall \ (x, a) \in \mathcal{X} \times \mathcal{A}$.*

The condition $\langle \mathbf{1}, \theta \rangle = 0$ ensures identifiability of θ^*. Now, we define a key quantity that captures learning complexity under the BTL preference model:

$$\kappa = \max_{x \in \mathcal{X}} \max_{a,a' \in \mathcal{A}} \max_{\theta \in \Theta} \frac{1}{\dot{\sigma}(\phi(x,a)^\top \theta - \phi(x,a')^\top \theta)} \ . \tag{7}$$

This parameter κ specifies difficulty in learning via the worst-case non-linearity in preference feedback. Note that we don't need the knowledge of κ beforehand. Next, we present the guarantee that our algorithm enjoys.

Theorem 3 (Sub-optimality gap of `APO`). *Let $\delta \in (0,1]$. Under Assumption 1, setting $\lambda = \frac{1}{4S^2(2+2S)^2}$ and $\gamma = O\left(S\sqrt{d \log(ST/d) + \log(T/\delta)}\right)$, the policy π_T returned by `APO`, with probability at least $1-\delta$, enjoys the suboptimality gap*

$$R(T) \le O\left(\gamma \sqrt{S \log\Big(1 + (T/\lambda\kappa d)\Big)\kappa d/T}\right) \ .$$

Comparison with lower bound and dependence on κ. Theorem 3 implies an $\widetilde{O}\big(d\sqrt{\kappa/T}\big)$ upper bound on the sub-optimality gap of `APO` policy. This matches the lower bound of Theorem 2 in parameter dimension d and number of samples T, implying optimal scaling w.r.t. these two terms (up to a log factor). There remains a gap in characterizing the optimal dependence on the non-linearity parameter κ, which, in the worst-case, can be exponential in the parameter norm S. In the logistic bandit literature, the state-of-the-art regret guarantee is (almost) κ-independent - the dependence is only in terms independent of T [27,13]. We believe the $\sqrt{\kappa}$ dependence is unavoidable in the RLHF setting as the sub-optimality gap is w.r.t. real-valued rewards $r^*(x,a) = \phi(x,a)^\top \theta^*$ instead of the sigmoid rewards $\sigma(\phi(x,a)^\top \theta^*)$ in logistic bandits. Given this, we conjecture that it could be possible to improve the lower bound of Theorem 2 to $\Omega\big(d\sqrt{\kappa/T}\big)$. We keep this as an interesting future direction.

Proof (Sketch). First, we quantify the error in estimating θ^* in the following lemma. This is obtained by using a novel inequality derived from the self-concordance property of the sigmoid function (i.e., $|\ddot{\sigma}| \le \dot{\sigma}$) and adapting the arguments from [13]. Proof of this lemma is deferred to Appendix C.

Lemma 1 (Estimation error at round t). *Let $\delta \in (0,1]$. Under the hypothesis of Theorem 3, with probability $\ge 1-\delta$, for some universal constant $C > 0$,*

$$\|\theta^* - \widehat{\theta}_t\|_{H_t(\widehat{\theta}_t)} \le C S^{3/2} \sqrt{d \log\left({}^{St}/_{d}\right) + \log\left({}^{t}/_{\delta}\right)} \ ,$$

where $\widehat{\theta}_t$ is the estimated reward parameter that minimizes the constrained log-loss (1) and $H_t(\widehat{\theta}_t)$ is the (scaled) design matrix (4) at round t.

The proof of Theorem 3 proceeds by upper bounding the sub-optimality gap for every context with the error in parameter estimation times an arm-dependent quantity. Specifically, for context x, let $z_T(x) = \phi(x, a^*(x)) - \phi(x, \pi_T(x))$ denotes the fetaure difference for the triplet $(x, a^*(x), \pi_T(x))$, where $a^*(x)$ is the optimal action at context x. Then, from (2), the sub-optimality gap for context x can be bounded as $z_T(x)^\top\theta^* \le z_T(x)^\top\theta^* - z_T(x)^\top\theta_T = \frac{1}{T}\sum_{t=1}^{T} z_T(x)^\top(\theta^* - \widehat{\theta}_t)$. Here, the first inequality is because $\phi(x, \pi_T(x))^\top\theta_T \ge \phi(x, a^*(x))^\top\theta_T$, which follows from definition of π_T, and so $z_T(x)^\top\theta_T \le 0$. Then, by Cauchy-Schwarz inequality, we get $z_T(x)^\top\theta^* \le \frac{1}{T}\sum_{t=1}^{T}\|z_T(x)\|_{H_t(\widehat{\theta}_t)^{-1}}\|\theta^* - \widehat{\theta}_t\|_{H_t(\widehat{\theta}_t)}$.

Now, we apply Lemma 1 to upper bound $\|\theta^* - \widehat{\theta}_t\|_{H_t(\widehat{\theta}_t)}$. Next, we note that $\|z_T(x)\|_{H_t(\widehat{\theta}_t)^{-1}} \le \|z_t\|_{H_t(\widehat{\theta}_t)^{-1}}$ by the design of our algorithm. To bound this, consider the regularized sample covariance matrix of feature differences, defined as $V_t = \sum_{s=1}^{t-1} z_s z_s^\top + \kappa\lambda\mathbf{I}_d$. Compare this with $H_t(\theta)$, which scales each rank-one component inside the sum by its variance given that the parameter is θ (see Eq. 4). A key relation between these two matrices is that $H_t(\theta) \succcurlyeq V_t/\kappa$. Using this, we upper bound $\|z_t\|_{H_t(\widehat{\theta}_t)^{-1}}$ by $\sqrt{\kappa}\|z_t\|_{V_t^{-1}}$. Finally, applying Elliptic Potential Lemma (Lemma 10) finishes the proof. Details are in Appendix C. □

Remark 2. To the best of our knowledge, [18] is the only work similar to ours with theoretical guarantees. Therefore, we highlight in detail the major differences. First, the algorithm design is entirely different as we choose both the actions for any context by maximizing the exploration bonus $b_t(x, a, a')$, while [18] chooses one action uniformly at random. This can be wasteful in practice as the choice of the second action is equally crucial in preference-based learning. Further, the context selection rule is entirely different. While we pick the context with the highest exploration bonus (Eq. 5), [18] uses an uncertainty estimate calculated via upper and lower confidence bounds of the rewards. Moreover, their suboptimality gap scales linearly with κ, while our guarantee only scales as $\sqrt{\kappa}$.

Next, [18] competes against the *Borda* winner, while we do so against the (stronger) *Condorcet* winner. Moreover, [18] assumes that the Borda function $g^*(x, a) = \mathbb{E}_{a' \sim \texttt{Unif}(\mathcal{A})}[\sigma(r^*(x, a) - r^*(x, a'))]$ lie in the same function space as the reward function $r^*(x, a) = \langle\theta^*, \phi(x, a)\rangle$. This assumption doesn't hold in general due to the non-linearity in σ and hence restricts the preference probabilities. The assumption holds trivially if each $\phi(x, a)$ is a one-hot vector $\mathbf{e}_{x,a}$, but it pushes θ^* to $|\mathcal{X}| \cdot |\mathcal{A}|$ dimensions and blows up the suboptimality gap significantly. We remove this restriction and provide the guarantee in dimension $d \ll |\mathcal{X}| \cdot |\mathcal{A}|$, by crucially exploiting properties of the sigmoid function.

Remark 3 (Extension to Direct Preference Optimization (DPO)). Another popular alignment algorithm DPO [22] does not train a reward model separately, rather it uses the log-probability $r_\theta(x, a) = \log\pi_\theta(a|x) - \log\pi_{\text{ref}}(a|x)$ as the reward, where $\theta \in \mathbb{R}^d$ parameterizes the policy to be learnt. For example, Softmax policies take the form $\pi_\theta(a|x) \propto \exp(f_\theta(x, a))$, where f_θ is a differentiable function. Now if we assume f_θ to be linear, then π_θ becomes a log-linear policy, i.e., $\log\pi_\theta(a|x) \propto \langle\theta, \phi(x, a)\rangle$, which eventually makes r_θ a (shifted) linear function. Hence, one would be able to apply our proposed approach to learn the policy parameter θ directly from the preference dataset $\mathcal{D}$.

4.2 Generalization Beyond BTL Model

In this section, we remove the assumption of the BTL preference model and assume access to a non-parametric function class

$$\mathcal{F} = \{f : \mathcal{X} \times \mathcal{A} \times \mathcal{A} \to [0,1] : f(x,a,a') + f(x,a',a) = 1\} ,$$

where $f(x,a,a')$ denotes the probability that action a wins over action a' (denoted by $a \succcurlyeq a'$) given context x when the preference function is f, i.e., $f(x,a,a') = \mathbb{P}[a \succcurlyeq a' | x, f]$. Note that in this case, there is no latent reward model, and hence, this is a strict generalization of the BTL model. Now, we assume that there is a realizable $f^* \in \mathcal{F}$ from which preferences are observed at each round t, i.e. $y_t \sim \texttt{Ber}(f^*(x_t, a_t, a'_t))$. Further, we assume a *Condorcet* winner at each context $x \in \mathcal{X}$ w.r.t. f^*, i.e. there is an action $a^*(x) \in \mathcal{A}$ such that $f^*(x, a^*(x), a) \geq {}^1\!/_2$ for all $a \in \mathcal{A}$. Accordingly, the sub-optimality gap of policy π_T is defined as

$$R(T) = \max_{x \in \mathcal{X}} f^*(x, a^*(x), \pi_T(x)) - {}^1\!/_2 .$$

In this setting, we propose a generalization of `APO`, namely, `APO-Gen` (Algorithm 3 in Appendix D). Similar to `APO`, it selects a context and a pair of actions at each round by maximizing an uncertainty score that depends on the complexity of the function class $\mathcal{F}$. However, unlike `APO`, (a) at each round, it prunes out sub-optimal actions for every context, and (b) after T rounds, the final policy is to sample an action uniformly from the remaining near-optimal actions for each context. `APO-Gen` enjoys the following guarantee (details, proof in Appendix D).

Theorem 4 (Suboptimality Gap of `APO-Gen`). *Let $\delta \in (0,1)$, and $\mathcal{N}(\mathcal{F})$ and $d_{\mathcal{E}}(\mathcal{F})$ be the covering number and Eluder dimension of $\mathcal{F}$ respectively. Then, with probability $\geq 1-\delta$, the policy returned by `APO-Gen` enjoys sub-optimality gap*

$$R(T) \leq \widetilde{O}\left(\sqrt{\log(\mathcal{N}(\mathcal{F})T/\delta) d_{\mathcal{E}}(\mathcal{F})/T}\right) ,$$

For the BTL preference model, we have $\log \mathcal{N}(\mathcal{F}) = O(d \log T)$ and $d_{\mathcal{E}}(\mathcal{F}) = O(\kappa^2 d \log T)$. Hence, we get an $\tilde{O}(\kappa d/\sqrt{T})$ sub-optimality gap for `APO-Gen`, which is $\sqrt{\kappa}$ factor loose than Theorem 3. This is because we crucially use self-concordance of the sigmoid function in Theorem 3 to shave this extra $\sqrt{\kappa}$ factor. Nevertheless, this result is general enough to subsume other preference models (e.g., Thurstone) beyond the BTL model.

5 Experiments

We first present a practical version of `APO`, which largely follows the former with minor changes adapted for the computationally efficient implementation required in large-scale experiments. Next, we present experimental results that demonstrate the efficacy of `APO` over random sampling (hereafter denoted by `Random`) and baselines [18,19]. Hyperparameter details are given in Appendix F. The experiment code can be found here.

Algorithm 2 APO (Practical version)

```
Require: Context-generation pairs $\mathcal{M} = \{(x,a,a')\}$, sample budget $T$, encoder $\phi$, SFT
    policy $\pi_{\mathrm{SFT}}$, log-loss $\mathcal{L}$, batch size $B$, uncertainty regularizer $\lambda > 0$, KL regularizer
    $\beta > 0$, learning rate $\eta > 0$. Initialize $V_1 = \lambda I, \widehat{\theta}_1 = 0, \mathcal{D} = \emptyset$.
1: for batch $t = 1, \dots, \lfloor T/B \rfloor$ do
2:     Set $b_t(x,a,a') = \|\phi(x,a) - \phi(x,a')\|_{V_t^{-1}}$ for each $(x,a,a') \in \mathcal{M}$. Set $\mathcal{M}_t = \emptyset$.
3:     for $j = 1, \dots, B$ do
4:         Pick $(x_{t,j}, a_{t,j}, a'_{t,j}) = \underset{(x,a,a') \in \mathcal{M} \setminus \mathcal{M}_t}{\mathrm{argmax}} \; b_t(x,a,a')$; Observe preference $y_{t,j}$.
5:         Update $\mathcal{M}_t \leftarrow \mathcal{M}_t \cup \{(x_{t,j}, a_{t,j}, a'_{t,j})\}$ and $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x_{t,j}, a_{t,j}, a'_{t,j}, y_{t,j})\}$.
6:     Update $\widehat{\theta}_{t+1} \leftarrow$ Gradient-step$(\mathcal{L}, \widehat{\theta}_t, \mathcal{D}, \eta)$.
7:     Update $V_{t+1} \leftarrow V_t + \sum_{j=1}^{B} z_{t,j} z_{t,j}^\top$, where $z_{t,j} = \phi(x_{t,j}, a_{t,j}) - \phi(x_{t,j}, a'_{t,j})$.
8: Set reward $\widehat{r}_T(x,a) = \phi(x,a)^\top \widehat{\theta}_{\lfloor T/B \rfloor + 1} \forall (x,a)$ and policy $\pi_T \leftarrow$ PPO$(\pi_{\mathrm{SFT}}, \widehat{r}_T, \beta)$.
```

5.1 Practical Version of APO for RLHF

In this practical version of APO (Algorithm 2), we access preference data in batches instead of being fully online. At the start of each batch t, we first compute the uncertainty $b_t(x,a,a')$ of each triplet (x,a,a') (Step 2). This is similar to Algorithm 1 except that here we compute the norm w.r.t. the inverted sample covariance matrix of feature differences V_t^{-1} instead of $H_t(\widehat{\theta}_t)^{-1}$ (Step 7). We do so since it is both compute and memory efficient for large scale experiments. To maximize exploration, only those B triplets (x,a,a') are sent for labeling in a batch that have the highest uncertainty $b_t(x,a,a')$, and those are stored in a buffer $\mathcal{D}$. At the end of each batch t, we update the parameter estimate $\widehat{\theta}_t$ via a black-box gradient-descend-based algorithm (e.g. Adam [11]) on the log-loss (1) over the dataset $\mathcal{D}$. Finally, after the budget T is exhausted, we first learn an estimate $\widehat{r}_T$ of the latent reward model r^*, and then align the policy via proximal policy optimization (PPO) [28], which takes as input the SFT policy π_{SFT}, the learnt reward model $\widehat{r}_T$ and a KL-regularizer β, and returns π_T.

5.2 Results on Controlled Sentiment Generation Task

In this experiment, we consider a user group that prefers positive sentiment completions for movie reviews in the IMDb dataset [16]. The goal is to output generations a that exhibit positive sentiment, catering to the user group's preferences for a given context x. For controlled evaluation, we generate preference pairs (a,a') utilizing a pre-trained sentiment classifier where $\mathbb{P}(\text{positive-sentiment} \mid x,a) > \mathbb{P}(\text{positive-sentiment} \mid x,a')$. We generate a total of 10000 preference samples (x,a,a') and use 4:1 train-test split. For the SFT policy, we fine-tune GPT-2 [21] on preferred reviews from the train set (8000 samples) and use this GPT-2 backbone for both reward learning and policy alignment.

For reward training, we adaptively select context and generation pairs from the train set. We use the feature representation $\phi(x,a)$, and estimate the uncertainty $b_t(x,a,a')$ for each (x,a,a') in the train set and select top-B samples

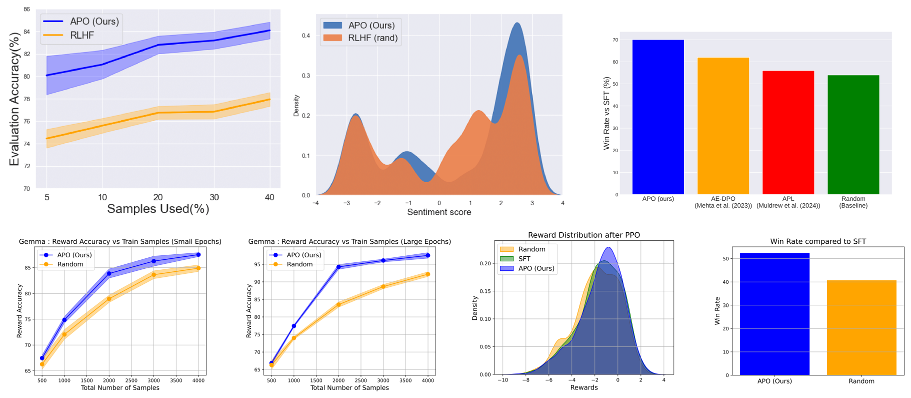

Fig. 1. Top Row: Controlled Sentiment Generation Task: Left: Evaluation accuracy of trained reward model vs. no. of samples (in %) comparing `APO` with `Random`. **Middle:** Sentiment score distribution of aligned policies trained on reward model learned with `APO` and on `Random`'s highest accuracy reward model. Generations by `APO`-trained reward is more shifted towards positive, showing better alignment than `Random`. **Right:** Win rates of `APO`, `AE-DPO` [18] and `APL` [19] and `Random` against SFT policy. `APO` outperforms `AE-DPO`, `APL` and `Random` by $72 : 62 : 56 : 54$ win rate. **Bottom Row: Single-turn Dialogue Task: Left and 2nd Left:** Evaluation accuracy of trained reward model vs. no. of samples comparing `APO` with `Random`, when the number of epochs is 5 (**Left**) and 20 (**2nd Left**). Evaluation accuracy of `APO` is higher than the `Random` in both cases. **2nd Right:** Reward distribution of `APO`-aligned, SFT and `Random`-aligned policies for generations on prompts in the test dataset. Clearly, `APO`'s alignment is better than `Random`. **Right:** Win rates of `APO` and `Random` aligned policies against SFT policy. `APO` outperforms `Random` by $55 : 40$ win rate.

to update the reward model. We repeat this process K times and return the final trained reward model. We evaluate the performance of the trained reward model against `Random` (where we select B samples randomly at every batch) on the test set of 2000 samples. Figure 1 (**Top Left**) shows the result: evaluation accuracy of the reward model learned by `APO` is much higher than the one learned via `Random` even when `APO`'s sample budget is only 5% of the data and `Random`'s is 40%.

For policy alignment, we align the SFT policy with respective trained reward models (via `APO` and `Random`) using `PPO` (step 8). To demonstrate the effectiveness of adaptive sampling, we use the reward model trained on a sample budget of only 10% for `APO`, while we use the highest accuracy reward model (corresponding to 40% samples) for `Random`. Similar to [22], the generations of aligned policies are evaluated against the ground truth reward r^* for positive sentiment, which is provided by the pre-trained sentiment classifier. From Figure 1 (**Top Middle**), it can be seen that the reward distribution of the generations of `APO` -aligned policy achieves a higher density of positive sentiment compared to that aligned by `Random`. Moreover, from Figure 1 (**Top Right**) it is evident that

APO outperforms APL [19], AE-DPO [18] and Random in terms of win-rate against the SFT policy demonstrating the efficiency of the proposed method.

5.3 Results on Single-turn Dialogue task

In this experiment, we use Anthropic-HH [2] preference dataset and instruction-tuned Gemma-2b [30] language model. We collect all the contexts with single-turn dialogues and split these into two sets in a 4:1 ratio. We put samples from the larger collection into three buckets based on reward difference between *chosen* and *rejected* responses using Mistral-7b reward model as latent reward r^*. These buckets contain data points that are progressively easier to classify: (B1) reward difference between -1 to 1, (B2) reward difference between 1 to 3 and (B3) reward difference of more than 3. Out of these three buckets, we take 4500 samples from (B1), 2500 from (B2), and 1000 from (B3) to carefully curate a collection of 8000 training samples. Such a collection (more samples taken from the buckets with a smaller reward difference and fewer samples from the bucket with a larger reward difference) highlights the importance of selecting prompts carefully to obtain useful information during reward training – randomly sampling contexts to collect feedback is more likely to hurt the performance in such a setting. For the test set, we sample 2000 data points from the smaller collection set aside.

Reward Evaluation. We compare the reward models learnt by APO and Random by computing the % of samples in the test set for which the models assign higher reward to the *chosen* responses than to the *rejected* responses. We study how this accuracy changes with the number of batches or epochs, keeping the sample budget the same (Fig. 1 (**Bottom Left**) for 5 epochs and Fig. 1 (**Bottom 2nd Left**) for 20 epochs). We observe that APO always outperforms Random. We also see that the reward accuracy increases with an increasing number of epochs for a given sample budget. We show results by varying training samples till 4000 as we want to demonstrate the effectiveness of APO under budget constraint.

Win Rate. Based on reward models learnt by APO and Random, we fine tune the SFT policy with PPO to obtain APO-policy and Random-policy respectively. Then we generate responses for contexts in the test set using APO, Random, and SFT policies, and get them evaluated by the Mistral-7b reward model. The reward distributions of these three policies are shown in Fig. 1 (**Bottom 2nd right**). It can be seen that APO has a higher density of positive rewards. Win rate of APO-policy and Random-policy against SFT policy is shown in Fig. 1 (**Bottom Right**), which shows that APO outperforms Random by a $55:40$ win rate.

6 Conclusion

In this work, we studied whether sampling prompts uniformly at random from a dataset to solicit feedback is sample efficient. We first showed that this method can suffer a constant suboptimality gap when aligning a language model policy with human preferences. Next, we characterized the sub-optimality gap lower bound for any active-learning algorithm with sample budget T and problem dimension d, showing it to be $\Omega(d/\sqrt{T})$. Then, we proposed an algorithm `APO`, which actively samples prompts to achieve an $\tilde{O}(d/\sqrt{T})$ sub-optimality gap. We also extended the results of `APO` to general function approximation to better capture modern-day RLHF training. Finally, we showed its efficacy over the random-sampling baseline on practical datasets. Although `APO`'s sub-optimality gap is minimax optimal in d and T, the optimal dependence on κ is unknown and seems to be an interesting future work.

References

1. Abeille, M., Faury, L., Calauzènes, C.: Instance-wise minimax-optimal algorithms for logistic bandits. In: International Conference on Artificial Intelligence and Statistics. pp. 3691–3699. PMLR (2021)
2. Bai, Y., et al.: Training a helpful and harmless assistant with reinforcement learning from human feedback (2022)
3. Bradley, R.A., Terry, M.E.: Rank analysis of incomplete block designs: I. the method of paired comparisons. Biometrika **39**(3/4), 324–345 (1952)
4. Carvalho Melo, L., Tigas, P., Abate, A., Gal, Y.: Deep bayesian active learning for preference modeling in large language models. Advances in Neural Information Processing Systems **37**, 118052–118085 (2024)
5. Chen, X., Zhong, H., Yang, Z., Wang, Z., Wang, L.: Human-in-the-loop: Provably efficient preference-based reinforcement learning with general function approximation. In: International Conference on Machine Learning. pp. 3773–3793 (2022)
6. Christiano, P.F., Leike, J., Brown, T., Martic, M., Legg, S., Amodei, D.: Deep reinforcement learning from human preferences. Advances in neural information processing systems **30** (2017)
7. Even-Dar, E., Mannor, S., Mansour, Y.: Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. Journal of Machine Learning Research **7**(39), 1079–1105 (2006)
8. Glaese, A., et al.: Improving alignment of dialogue agents via targeted human judgements. arXiv preprint arXiv:2209.14375 (2022)
9. Hazan, E., et al.: Introduction to online convex optimization. Foundations and Trends® in Optimization **2**(3-4), 157–325 (2016)
10. Ji, K., He, J., Gu, Q.: Reinforcement learning from human feedback with active queries. arXiv preprint arXiv:2402.09401 (2024)
11. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR). San Diego, CA, USA (2015)
12. Lattimore, T., Szepesvári, C.: Bandit algorithms. Cambridge University Press (2020)
13. Lee, J., Yun, S.Y., Jun, K.S.: Improved regret bounds of (multinomial) logistic bandits via regret-to-confidence-set conversion. In: International Conference on Artificial Intelligence and Statistics. pp. 4474–4482. PMLR (2024)
14. Lee, K., Smith, L.M., Abbeel, P.: Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In: Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 6152–6163. PMLR (2021)
15. Luce, R.D.: Individual choice behavior. John Wiley (1959)

16. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. pp. 142–150. Association for Computational Linguistics (2011)
17. Maiti, A., Boczar, R., Jamieson, K., Ratliff, L.: Near-optimal pure exploration in matrix games: A generalization of stochastic bandits and dueling bandits. In: Proceedings of The 27th International Conference on Artificial Intelligence and Statistics. pp. 2602–2610 (2024)
18. Mehta, V., Das, V., Neopane, O., Dai, Y., Bogunovic, I., Schneider, J., Neiswanger, W.: Sample efficient reinforcement learning from human feedback via active exploration. arXiv preprint arXiv:2312.00267 (2023)
19. Muldrew, W., Hayes, P., Zhang, M., Barber, D.: Active preference learning for large language models. In: Proceedings of the 41st International Conference on Machine Learning. pp. 36577–36590. PMLR (21–27 Jul 2024)
20. Ouyang, L., et al.: Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems (2022)
21. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)
22. Rafailov, R., Sharma, A., Mitchell, E., Manning, C.D., Ermon, S., Finn, C.: Direct preference optimization: Your language model is secretly a reward model. In: Thirty-seventh Conference on Neural Information Processing Systems (2023)
23. Ray Chowdhury, S., Kini, A., Natarajan, N.: Provably robust DPO: Aligning language models with noisy feedback. In: Proceedings of the 41st International Conference on Machine Learning. pp. 42258–42274. PMLR (21–27 Jul 2024)
24. Sadigh, D., Dragan, A.D., Sastry, S.S., Seshia, S.A.: Active preference-based learning of reward functions. In: Robotics: Science and Systems (2017)
25. Saha, A.: Optimal algorithms for stochastic contextual preference bandits. Advances in Neural Information Processing Systems **34**, 30050–30062 (2021)
26. Saha, A., Pacchiano, A., Lee, J.: Dueling RL: Reinforcement learning with trajectory preferences. In: Proceedings of The 26th International Conference on Artificial Intelligence and Statistics. pp. 6263–6289. PMLR (25–27 Apr 2023)
27. Sawarni, A., Das, N., Barman, S., Sinha, G.: Generalized linear bandits with limited adaptivity. Advances in Neural Information Processing Systems (2024)
28. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
29. Stiennon, N., et al.: Learning to summarize with human feedback. Advances in Neural Information Processing Systems **33**, 3008–3021 (2020)
30. Team, G., et al.: Gemma: Open models based on gemini research and technology. arXiv preprint arXiv:2403.08295 (2024)
31. Zhan, W., Uehara, M., Sun, W., Lee, J.D.: How to query human feedback efficiently in rl? ArXiv **abs/2305.18505** (2023)
32. Zhu, B., Jordan, M., Jiao, J.: Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In: Proceedings of the 40th International Conference on Machine Learning. pp. 43037–43067. PMLR (2023)