HYBRID PREFERENCE OPTIMIZATION FOR ALIGNMENT: PROVABLY FASTER CONVERGENCE RATES BY COMBINING OFFLINE PREFERENCES WITH ONLINE EXPLORATION

Avinandan Bose
University of Washington
avibose@cs.washington.edu

Zhihan Xiong
University of Washington
zhihanx@cs.washington.edu

Aadirupa Saha
Apple
aadirupa@apple.com

Simon Shaolei Du
University of Washington
ssdu@cs.washington.edu

Maryam Fazel
University of Washington
mfazel@uw.edu

ABSTRACT

Reinforcement Learning from Human Feedback (RLHF) is currently the leading approach for aligning large language models with human preferences. Typically, these models rely on extensive offline preference datasets for training. However, offline algorithms impose strict concentrability requirements, which are often difficult to satisfy. On the other hand, while online algorithms can avoid the concentrability issue, pure online exploration could be expensive due to the active preference query cost and real-time implementation overhead. In this paper, we propose a novel approach: Hybrid Preference Optimization (HPO) which combines online exploration with existing offline preferences by relaxing the stringent concentrability conditions for offline exploration, as well as significantly improving the sample efficiency for its online counterpart. We give the first provably optimal theoretical bound for Hybrid RLHF with preference feedback, providing sample complexity bounds for policy optimization with matching lower bounds. Our results yield improved sample efficiency of hybrid RLHF over pure offline and online exploration.

1 INTRODUCTION

Reinforcement Learning from Human Feedback (RLHF) stands out as the primary method for aligning large language models with human preferences (Christiano et al., 2017; Bai et al., 2022; Ouyang et al., 2022). Instead of starting from scratch with unsupervised training on extensive datasets, RLHF aligns pre-trained models using labeled human preferences on pairs of responses, offering a statistically lightweight approach to making language models more human-like. While labeling response pairs is easier than generating new responses, the volume of these pairs is critical for effective alignment. A large dataset is needed to ensure broad coverage of linguistic nuances, reduce the impact of noisy human feedback, and provide enough statistical power for the model to generalize well. Although labeling individual pairs is simpler, scaling this process can still become resource-intensive, making the volume of response pairs a key factor in successful model alignment. In the light of this, recently a theoretical question of interest has arisen: *How can algorithms be designed to be sample-efficient during this alignment phase?*

Two main approaches have emerged in addressing this question: online RLHF and offline RLHF. Online methods (Xie et al., 2024; Cen et al., 2024; Zhang et al., 2024) have interactive access to human feedback or leverage a more powerful language model to explore diverse and novel responses beyond what the pre-trained model can provide. Online exploration, though theoretically sample-efficient, is costly because it requires frequent human feedback, real-time updates to large language models, and continuous deployment to generate novel responses. These frequent model updates and the need for real-time interaction with human annotators make the process resource-intensive and time-consuming, presenting significant practical challenges in terms of both scalability and cost.

In contrast, offline RLHF relies on large, readily available sets of labeled response pairs to align the language model. While these datasets are easier to collect and use, achieving provable guarantees on the model's optimality requires stringent conditions—specifically, that the dataset is close to data generated by an optimally-aligned model. Offline datasets, however, are static and often lack the diversity and novel responses that online exploration provides. This limitation, coupled with the assumption that the dataset contains enough near-optimal examples, restricts the model's ability to generalize effectively, potentially leading to suboptimal alignment.

This prompts the question: *Can we combine offline preference data while relaxing these stringent conditions to significantly reduce the number of samples required in online exploration, thereby addressing its practical costs effectively?*

In this paper, we answer this question affirmatively.

	Category	Upper Bound	Lower Bound	Scalable	Model-free
Xie et al. (2021)	Traditional RL	✓	✓	✗	✗
Tan et al. (2024)	Traditional RL	✓	✓	✗	✗
Xiong et al. (2024)	RLHF	✓	✗	✗	✓
Gao et al. (2024)	RLHF	✓	✗	✓	✓
Chang et al. (2024)	RLHF	✓	✗	✓	✓
Our work	RLHF	✓	✓	✓	✓

Table 1: A comparison with lines of work closest to ours.

Here, we list our technical contributions:

- We introduce the first Hybrid RLHF algorithm that is both provably efficient and practical for implementation. In contrast, previous discussions of hybrid training in Xiong et al. (2024), Gao et al. (2024), and Chang et al. (2024) either lack practical applicability or do not provide a theoretical analysis of the advantages of hybrid training. See Table 1 for a detailed comparison.
- Our algorithm demonstrates superior sample complexity compared to both online and offline RLHF approaches (Theorem 1). Specifically, in the case of linear MDPs, our method surpasses the limitations imposed by the lower bounds of purely online and offline training (Theorems 2 and 3).

2 Related Work

DPO-related RLHF. The Direct Preference Optimization (DPO) algorithm, first introduced by Rafailov et al. (2023), has gained considerable attention due to its effectiveness and simplicity. Since its introduction, various DPO variants have been proposed, each with unique enhancements aimed at improving performance. Several of these methods focus on modifying the loss function, such as SLiC (Zhao et al., 2023), RSO (Liu et al., 2023), P3O (Wu et al., 2024), PCO (Xu et al., 2023), and SimPO (Meng et al., 2024). Other approaches take a broader view by incorporating general preference modeling, integrating social choice theory, and introducing algorithms like Nash-MD (Munos et al., 2023), DNO (Rosset et al., 2024), KTO (Ethayarajh et al., 2024), and IPO (Azar et al., 2024).

DPO's popularity has also captured the interest of the theoretical research community. Xiong et al. (2024) explored the benefits of online iterative training with an offline warm-up phase, while Ye et al. (2024) extended this analysis from a reward-based framework to a general preference oracle setting. Both studies, however, limit their analysis to pure bandit environments. Very recently, the online convergence of DPO is more carefully analyzed by Shi et al. (2024). Meanwhile, Rafailov et al. (2024) further reinterpreted DPO through the lens of implicit $Q^\star$ estimation within KL-constrained deterministic token-level Markov Decision Processes (MDPs). Gao et al. (2024) introduced a study on relative reward modeling, while recent work by Cen et al. (2024) and Xie et al. (2024) analyzed the sample efficiency of combining DPO with optimistic exploration. This research has also inspired the development of the online exploration component in our Hybrid Preference Optimization (HPO) algorithm.

Despite these advancements, none of these has deep dive into a hybrid training setting. Cen et al. (2024) and Ye et al. (2024) have discussed offline and online learning separately, but neither has explored the potential benefits of combining the two. Xiong et al. (2024) addressed hybrid training but relied on impractical assumptions, such as coverage over the optimal policy $\pi^\star$, and failed to provide a quantitative characterization of hybrid RLHF compared to purely online or offline approaches. Similarly, while Gao et al. (2024) mentioned hybrid training in their work, they did not delve deeply into the topic, leaving an open area for further investigation. Finally, although the same algorithm name *Hybrid Preference Optimization* is also used in Song et al. (2024), that version is essentially an offline DPO regularized by unlabeled online data, which is very different from our approach of enhancing online exploration through additional offline data.

Other Theoretical Work in RLHF Other theoretical work on RLHF primarily centers around learning reward models and analyzing sample complexities for specific classes of Markov Decision Processes (MDPs). Among them, Zhu et al. (2023) and Zhu et al. (2024) have focused on learning various reward models, while others, including Du et al. (2024), Zhan et al. (2023), and Wu and Sun (2023), have addressed on sample complexities under tabular or linear function approximations. Some studies, such as those by Chen et al. (2022), Wang et al. (2023), and Chang et al. (2024), also explore the use of general function approximation methods. However, these algorithms are often designed around function classes with certain complexity measures, making them less applicable in practical settings. Besides, Nika et al. (2024) provides a detailed theoretical comparison between DPO-related RLHF and reward-learning-related RLHF. Among these studies, Chang et al. (2024) provides a more detailed discussion on hybrid training in RLHF. Despite this, their work shares similar limitations to Xiong et al. (2024), particularly in terms of too strong assumptions on single-concentrability coefficients.

Theory of Hybrid RL. Kalashnikov et al. (2018) empirically demonstrate that in robot manipulation, a small amount of online fine-tunning can largely improve the performance of offline training. Following this, much of the theoretical work on hybrid RL has focused on quantitatively analyzing the advantages of combining online exploration with an initial offline dataset. Key studies by Xie et al. (2021), Song et al. (2022), and Amortila et al. (2024) have explored these benefits, often under the assumption of certain concentrability conditions related to the behavior policy. However, recent research by Tan et al. (2024) removes this requirement, offering an approach that does not impose explicit constraints on the quality of the behavior policy. Building on this line of research, our work extends the benefits of hybrid RL to the RLHF framework. Importantly, we adopt the same philosophy as Tan et al. (2024) by not imposing additional requirements on the offline dataset, allowing for more flexible and practical applications in RLHF scenarios.

3 Preliminaries

Our study of RLHF is under the general reinforcement learning setup similar to (Xie et al., 2024) which is a strict generalization of the token-level MDP proposed in (Rafailov et al., 2024).

3.1 Reinforcement Learning from Human Feedback

We consider an episodic finite horizon markov decision process characterized by $\mathcal{M} = (H, \mathcal{S}, \mathcal{A}, P, r, \rho)$ where H is the time horizon, $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, $P : \mathcal{S} \times \mathcal{A} \mapsto \Delta(\mathcal{S})$ denotes the transition function, $r : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ denotes the reward function, and $\rho \in \Delta(\mathcal{S})$ denotes the initial state distribution. A randomized policy $\pi : \mathcal{S} \mapsto \Delta(\mathcal{A})$ generates a trajectory $\tau = \{(s_1, a_1), \ldots, (s_H, a_H)\}$ and rewards $r_1, \ldots, r_H$ through the following procedure. In particular, the initial state is sampled as $s_1 \sim \rho$, and for all subsequent time steps $h \in [H]$, actions are sampled as $a_h \sim \pi(s_h)$, reward is received as $r_h = r(s_h, a_h)$, and $s_{h+1} \sim P(s_h, a_h)$. We use shorthand notations $\mathbb{E}_\pi[\cdot]$ and $\mathbb{P}_\pi[\cdot]$ to denote the expectation and probability of quantities of interest under this process induced by the policy π. We assume the total reward is non-negative and bounded, i.e. $\sum_{h\in[H]} r_h \in [0, R_{\max}]$. For compact notation at the trajectory level, we denote $r(\tau) = \sum_{h\in[H]} r(s_h, a_h)$ and $\pi(\tau) = \prod_{h\in[H]} \pi(a_h|s_h)$. The expected reward under policy π is defined as $J(\pi) = \mathbb{E}_{\tau\sim\pi}[r(\tau)]$.

Preference Model In RLHF, the reward signal from a trajectory $(r_1, \ldots, r_H)$ is generally unobservable. Instead, the feedback from the environment is provided through a preference oracle. In particular, given a pair of trajectories $(\tau, \tilde{\tau})$, both having initial state s_1, we follow the standard assumptions as in (Christiano et al., 2017; Rafailov et al., 2023; 2024), that the probability of τ being preferred over $\tilde{\tau}$ follows the Bradley-Terry Model (Bradley and Terry, 1952), which is defined as

$$\mathbb{P}(\tau \succ \tilde{\tau}|s_1) = \frac{\exp\left(r(\tau)\right)}{\exp\left(r(\tau)\right) + \exp\left(r(\tilde{\tau})\right)}. \tag{1}$$

Typically, we have access to a pre-trained policy π_{ref}, and the goal of RLHF is to find the policy that can maximize $J(\pi)$ while staying not too far away from π_{ref}. That is, with regularization parameter $\beta > 0$, we aim to maximize the following objective:

$$\begin{aligned} J_\beta(\pi) &= J(\pi) - \beta \mathbb{E}_\pi \left[\sum_{h \in [H]} D_{KL}(\pi(\cdot|s_h)||\pi_{\text{ref}}(\cdot|s_h)) \right] \\ &= \mathbb{E}_\pi \left[r(\tau) - \beta \log \frac{\pi(\tau)}{\pi_{\text{ref}}(\tau)} \right]. \end{aligned} \tag{2}$$

Here, we denote the optimal policy of some policy class Π as $\pi^\star_\beta = \arg\max_{\pi \in \Pi} J_\beta(\pi)$.

Token-level MDPs. Given our primary focus on language models, we are particularly interested in the *token-level MDP* framework, as formulated and studied by Rafailov et al. (2024). In this context, a language model typically generates a sequence of responses from an initial prompt. The initial prompt is treated as the initial state, $s_1 \sim \rho$, and each generated token, $a_h \in \mathcal{A}$, is considered an action, where $\mathcal{A}$ represents the vocabulary. Under this formulation, the state at step h is represented as $s_h = (s_1, a_1, \ldots, a_{h-1})$, a concatenation of the initial prompt and all tokens generated before step h. The state at last step, s_H, can then be interpreted as the model's complete response to the initial prompt, including the prompt itself.

Clearly, from this formulation, the transition is deterministic from s_h to s_{h+1} given action a_h since it is simply a concatenation. This is summarized as *Deterministic Contextual MDPs* (DCMDPs) in Xie et al. (2024), which proposes that the optimal policy $\pi^\star_\beta$ of the objective equation 2 in a DCMDP satisfies

$$\beta \log \frac{\pi^\star_\beta(\tau)}{\pi_{\text{ref}}(\tau)} = r(\tau) - V^\star_\beta(s_1), \quad \forall \tau, \tag{3}$$

where $V^\star_\beta = \max_{\pi \in \Pi} J_\beta(\pi)$. More details of DCMDP can be found in Xie et al. (2024).

To optimize objective equation 2, two primary approaches are available: offline RLHF and online RLHF. The choice between these methods depends on the specific requirements of the application and we will discuss both approaches in details in the following sections.

3.2 Offline, Online and Hybrid RLHF

Offline RLHF. Offline RLHF methods are restricted to the preferences captured by the reference model π_{ref} and the offline preference dataset $\mathcal{D}_{\text{off}}$. Typically the offline preference dataset is a set of labelled pairs $\mathcal{D}_{\text{off}} = \{(\tau^{(i)}_+, \tau^{(i)}_-)\}_{i \in [N_{\text{off}}]}$. To learn a policy from this offline dataset, one of the most popular approaches is Direct Preference Optimization (DPO) introduced in Rafailov et al. (2023), which also serves as a starting point of our work. DPO is motivated by a closed-form solution for the policy that optimizes the KL-regularized objective in Eq. equation 2, and condenses the two-step process above into a single policy optimization objective, removing the need for reward function estimation. Concretely, DPO solves the following problem:

$$\widehat{\pi} = \arg\min_{\pi \in \Pi} \sum_{(\tau_+, \tau_-) \in \mathcal{D}_{\text{off}}} -\log\left[\sigma\left(\beta z(\tau_+, \tau_-, \pi_{\text{ref}}, \pi)\right)\right], \tag{4}$$

where $z(\tau_+, \tau_-, \pi_{\text{ref}}, \pi) = \log \frac{\pi(\tau_+)}{\pi_{\text{ref}}(\tau_+)} - \log \frac{\pi(\tau_-)}{\pi_{\text{ref}}(\tau_-)}$, $\sigma(x) = \frac{\exp(x)}{1+\exp(x)}$ is the sigmoid function and Π is some user-specified policy class such as parameterized neural networks.

Online RLHF Although achieving great empirical success, offline DPO is inherently limited by the support of reference policy π_{ref} and offline dataset $\mathcal{D}_{\text{off}}$. Therefore, people turn more attention into online DPO, which was first theoretically studied in Xiong et al. (2024). Specifically, online RLHF collects preference feedback from pairs of responses generated by the language model sequentially updated during training. Recently, Xie et al. (2024) analyzed the necessity of deliberate exploration in online DPO and proposed XPO as a combination of DPO and optimistic exploration. In particular, the original DPO loss is regularized by $\alpha \sum_{i=1}^{t} \log \pi(\widetilde{\tau}^{(i)})$ in XPO to encourage the policy to be more diverse, where $\alpha > 0$ is the regularization strength and $\widetilde{\tau}^{(i)}$ is the trajectory generated by π_{ref}. It was analyzed under the framework of DCMDP and achieved $\widetilde{O}(1/\sqrt{T})$ optimality gap under appropriate assumptions, where T represents the number of queried preference labels. Concurrently, similar theoretical guarantee was also obtained by Cen et al. (2024).

Hybrid RLHF Although the efficiency of online RLHF is justified both theoretically and experimentally, in application, real-time query of labels can be expensive while a large amount of offline preference dataset can often be easily and cheaply obtained. Therefore, it is natural to consider a hybrid training scheme which can potentially combine the advantage of both offline and online RLHF. From a theoretical perspective, the sample complexity of pure online RLHF depends solely on the coverage provided by π_{ref}, meaning that the larger the coverage provided by π_{ref}, the smaller the space to search for online RLHF, thereby requiring fewer samples. However, pure online RLHF neglects the coverage provided by the offline preference dataset which can significantly reduce the search space. Therefore, we propose to use online feedback to explore trajectories that are not supported in π_{ref} while still taking advantage of the available offline dataset to speed up exploration. Informally, even starting with an offline dataset on which any offline algorithm may give arbitrarily poor policy, our proposed algorithm HPO will be able to use it to significantly reduce the sample complexity, as compared to XPO.

4 Hybrid Preference Optimization

We now present our algorithm Hybrid Preference Optimization (HPO), which integrates the available offline dataset with online exploration. This hybrid approach leverages the strengths of both offline data and online feedback, aiming to enhance the learning process and achieve a near optimal policy using fewer samples than pure online and pure offline methods.

4.1 Algorithm Description

We present Hybrid Preference Optimization (HPO) in Algorithm 1. Given a policy class Π, an offline dataset $\mathcal{D}_{\text{off}}$, the algorithm proceeds through a series of steps over a predefined number of iterations T. The algorithm begins with 2 policies $\pi^{(1)}, \tilde{\pi}^{(1)}$ and iteratively updates these policies as described next.

At each step $t \in [T]$ of the online exploration, a context is sampled as $s_1^{(t)} \sim \rho$. The algorithm generates 2 responses: one sampled from the current policy $\tau_t \sim \pi^{(t)}(\cdot|s_1^{(t)})$, and one sampled from the model $\tilde{\tau}_t \sim \tilde{\pi}^{(t)}(\cdot|s_1^{(t)})$. This response pair is then labelled as $(\tau_+^{(t)}, \tau_-^{(t)})$ based on human preference feedback and added to the online buffer $\mathcal{D}_{\text{on}}^{(t)} = \mathcal{D}_{\text{on}}^{(t)} \cup (\tau_+^{(t)}, \tau_-^{(t)})$.

A key parameter in HPO is $\gamma \in \mathbb{N}$, which determines the number of labeled pairs drawn from the offline dataset $\mathcal{D}_{\text{off}}^{(t)}$. The hybrid preference dataset at time t is the union of the online buffer and γ pairs sampled from the offline dataset $\mathcal{D}_{\text{hyb}}^{(t)} = \mathcal{D}_{\text{on}}^{(t)} \cup \mathcal{D}_{\text{off}}^{(t)}$. An optimism dataset $\mathcal{D}_{\text{opt}}^{(t)}$ is built by sampling $t + \gamma$ samples from $\tilde{\pi}^{(t)}$.

Now, we update the policy via an optimistic variant of DPO, and similar to XPO as in Eq. equation 5. Here α is an optimism parameter, and the first term in Eq. equation 5 tries to encourage the policy to explore more diverse responses compared to those generated by $\tilde{\pi}^{(t)}$ and captured by the samples in $\mathcal{D}_{\text{opt}}^{(t)}$. The second term in Eq. equation 5 is the DPO objective on the hybrid preference dataset $\mathcal{D}_{\text{hyb}}^{(t)}$ which aligns the policy to the human preferences.

Algorithm 1 Hybrid Preference Optimization

Input: Offline dataset $\mathcal{D}_{\text{off}}$ of size N_{off}, sampling strategy π_{samp}, hyper-parameters, $\alpha \in \mathbb{R}_+$, $\gamma \in \mathbb{N}$.
Initialize: $\pi_t, \tilde{\pi}_t \leftarrow \pi_{\text{ref}}$, $\mathcal{D}^0_{\text{on}} = \varnothing$.
for $t = 1, \ldots, T$ **do**
  Generate response pair: $s_1^{(t)} \sim \rho$; $\tau^{(t)} \sim \pi^{(t)}(\cdot|s_1^{(t)})$, $\tilde{\tau}^{(t)} \sim \tilde{\pi}_t(\cdot|s_1^{(t)})$.
  Label with Preference: Label $(\tau_t, \tilde{\tau}_t)$ as (τ_t^+, τ_t^-) based on preference feedback.
  Update Online Dataset: $\mathcal{D}_{\text{on}}^{(t)} \leftarrow \mathcal{D}_{\text{on}}^{(t-1)} \cup (\tau_+^{(t)}, \tau_-^{(t)})$.
  Update Offline dataset minibatch: Sample γ pairs from $\mathcal{D}_{\text{off}}$ uniformly randomly with replacement $\mathcal{D}_{\text{off}}^{(t)}$.
  Update Hybrid dataset: $\mathcal{D}_{\text{hyb}}^{(t)} \leftarrow \mathcal{D}_{\text{on}}^{(t)} \cup \mathcal{D}_{\text{off}}^{(t)}$.
  Update Optimism Dataset: Compute $\mathcal{D}_{\text{opt}}^{(t)}$ of $t + \gamma$ samples from $\tilde{\pi}_t$.
  Update Policy: Update π_t to maximize likelihood of preferences seen so far + regularization terms

$$\pi^{(t+1)} = \arg\min_{\pi \in \Pi} \Bigg[\alpha \sum_{\tau \in \mathcal{D}_{\text{opt}}^{(t)}} \log \pi(\tau) - \sum_{(\tau^+, \tau^-) \in \mathcal{D}_{\text{hyb}}^{(t)}} \log \left[\sigma(\beta \log \frac{\pi(\tau^+)}{\pi_{\text{ref}}(\tau^+)} - \beta \log \frac{\pi(\tau^-)}{\pi_{\text{ref}}(\tau^-)}) \right] \Bigg]. \tag{5}$$

  Update sampling policy: $\tilde{\pi}_{t+1} \leftarrow \pi_{\text{samp}}(\pi^{(1)}, \ldots, \pi^{(t+1)})$.
end for
Return $\widehat{\pi} = \text{argmax}_{\pi \in \{\pi^{(1)}, \ldots, \pi^{(T)}\}} J_\beta(\pi)$.

If $\gamma = 0$, the algorithm is exactly the same as the XPO in (Xie et al., 2024) and it discards all the available offline data during training. We will later show in Theorem 1 how the choice of γ shows up in our sample complexity bounds. Intuitively the value of γ, biases the policy to the behavior of the offline dataset. A too large γ will put a large emphasis on the offline dataset and may hinder exploration, while a too small γ will fail to effectively utilize the available information in the offline dataset. Thus, γ needs to be chosen in accordance to the online exploration budget T.

Finally, we can see that HPO is very friendly to application as it is rooted from the vanilla DPO and the extra regularization term and sampling procedure can be easily implemented.

4.2 Assumptions

To establish sample complexity guarantees for HPO, we adopt standard statistical assumptions. The first of these assumptions requires that the policy class Π be sufficiently expressive to represent the optimal KL-regularized policy.

Assumption 1 (Policy realizability)**.** The policy class Π contains the optimal policy, i.e., $\pi_\beta^\star \in \Pi$.

Policy realizability is a standard assumption for sample-efficient reinforcement learning which can be found in prior works such (Agarwal et al., 2019; Lattimore and Szepesvári, 2020; Foster and Rakhlin, 2023). It is equivalent to a form of reward/value realizability as discussed in (Xie et al., 2024). In our context, Π will typically correspond to a class of language models with fixed architecture and variable weights.

Next, we make a regularity assumption on the policies in Π (Rosset et al., 2024; Xie et al., 2024).

Assumption 2 (Bounded density ratios)**.** (Xie et al., 2024) For all $\pi \in \Pi$ and trajectories $\tau = (s_1, a_1), \ldots, (s_H, a_H)$, it holds

$$\left| \log \left(\frac{\pi(\tau)}{\pi_{\text{ref}}(\tau)} \right) \right| \leq \frac{V_{\max}}{\beta}. \tag{6}$$

As discussed in (Xie et al., 2024), $V_{\max}$ is measurable and controllable in practice. Specifically for log-linear policies where $\pi(a \mid s) \propto \exp(f(s,a)/\beta)$ with some linear function $f(s,a)$, it can be shown that $V_{\max} \lesssim R_{\max}$ (recall that $R_{\max}$ is the range of reward).

4.3 Model Complexity Measures

To measure the algorithm's convergence rate towards an optimal policy, we introduce an exploration criterion that limits how frequently the algorithm encounters novel samples where it cannot effectively distinguish between the impact of π and π_{ref} on the objective.

First, we define a preference-based analogue of the *Sequential Extrapolation Coefficient* (SEC)[1] from (Xie et al., 2023) that corresponds to the hybrid RLHF case. Particularly, define

$$g^{(\pi)}(\tau, \tilde{\tau}) = \left[\beta \log \frac{\pi(\tau)}{\pi_{\mathrm{ref}}(\tau)} - r(\tau) - \beta \log \frac{\pi(\tilde{\tau})}{\pi_{\mathrm{ref}}(\tilde{\tau})} + r(\tilde{\tau})\right]$$

as a measure of the difference in the objective from Eq. equation 2 for an arbitrary policy $\pi \in \Pi$ and two arbitrary trajectories τ and $\tilde{\tau}$.

For a pair of policies π and $\tilde{\pi}$, we define $\pi \otimes \tilde{\pi}$ as the joint policy that, given s_1, samples $\tau \sim \pi \mid s_1$ and $\tilde{\tau} \sim \tilde{\pi} \mid s_1$. We write $(\tau, \tilde{\tau}) \sim \pi \otimes \tilde{\pi} \mid s_1$ as a shorthand for this process. We further let $\tilde{\pi}^{(t)} = \pi_{\mathsf{samp}}(\pi^{(1)}, \ldots, \pi^{(t)})$, and $\mu^{(t)} = \frac{1}{t-1}\sum_{i=1}^{t-1} \pi^{(i)} \otimes \tilde{\pi}^{(i)}$, with the convention that $\mu^{(1)}$ is arbitrary and π_{samp} is some sampling strategy such as $\mathsf{unif}(\cdot)$. We first define the online sequential exploration coefficient for online RLHF (Xie et al., 2024).

Definition 4.1. (Xie et al., 2024) Given a policy class Π, reference policy π_{ref}, sampling strategy π_{samp}, entropy regularization parameter $\beta > 0$, online exploration budget $T \in \mathbb{N}$, we define the sequential exploration coefficient (SEC) as:

$$\mathrm{SEC}_{\mathrm{RLHF}}(\Pi, T, \beta, \pi_{\mathrm{samp}}) = \sup_{\pi^{(1)},\ldots,\pi^{(T)} \in \Pi} \sum_{t=1}^{T} \frac{\left(\mathbb{E}_{\substack{s_1 \sim \rho, \\ \tau \sim \pi^{(t)} \mid s_1, \\ \tilde{\tau} \sim \tilde{\pi}^{(t-1)} \mid s_1}} [g^{(\pi^{(t)})}(\tau, \tilde{\tau})]\right)^2}{V_{\max}^2 \vee \left[(t-1) \cdot \mathbb{E}_{s_1 \sim \rho, (\tau,\tilde{\tau}) \sim \mu^{(t)} \mid s_1} [\left(g^{(\pi^{(t)})}(\tau, \tilde{\tau})\right)^2]\right]}. \tag{7}$$

Next, we introduce another quantity to capture the coverage of the offline dataset in terms of any arbitrary policy $\pi \in \Pi$:

$$C_{\mathrm{off}}^{(\pi)} = \frac{1}{N_{\mathrm{off}}} \sum_{(\tau_+, \tau_-) \in \mathcal{D}_{\mathrm{off}}} \left(g^{(\pi)}(\tau_+, \tau_-)\right)^2. \tag{8}$$

Below we define the SEC coefficient in the Hybrid RLHF case:

Definition 4.2. Given a policy class Π, reference policy π_{ref}, offline preference dataset $\mathcal{D}_{\mathrm{off}}$, offline sampling parameter $\gamma \in \mathbb{N}$, sampling strategy π_{samp}, entropy regularization parameter $\beta > 0$, online exploration budget $T \in \mathbb{N}$, we define the sequential exploration coefficient (SEC) as:

$$\mathrm{SEC}_{\mathrm{HybRLHF}}(\Pi, T, \beta, \pi_{\mathrm{samp}}; \gamma, \mathcal{D}_{\mathrm{off}}) = \sup_{\pi^{(1)},\ldots,\pi^{(T)} \in \Pi} \sum_{t=1}^{T} \frac{\left(\mathbb{E}_{\substack{s_1 \sim \rho, \\ \tau \sim \pi^{(t)} \mid s_1, \\ \tilde{\tau} \sim \tilde{\pi}^{(t-1)} \mid s_1}} [g^{(\pi^{(t)})}(\tau, \tilde{\tau})]\right)^2}{V_{\max}^2 \vee \left[(t-1) \cdot \mathbb{E}_{s_1 \sim \rho, (\tau,\tilde{\tau}) \sim \mu^{(t)} \mid s_1} [(\widetilde{g}_{\mathrm{off}}^{(\pi^{(t)})})^2]\right]}, \tag{9}$$

where $(\widetilde{g}_{\mathrm{off}}^{(\pi)})^2 = \left(g^{(\pi)}(\tau, \tilde{\tau})\right)^2 + \gamma \cdot C_{\mathrm{off}}^{(\pi)}$.

[1] This is also known as an Eluder coefficient or decoupling coefficient (Zhong et al., 2022; Ye et al., 2024).

Remark. Note that because of the additional quantity $C_{\text{off}}^{(\pi)} \geq 0$, for all $\gamma \in \mathbb{N}$ and any non-empty $\mathcal{D}_{\text{off}}$, we have

$$\text{SEC}_{\text{HybRLHF}}(\Pi, T, \beta, \pi_{\text{samp}}; \gamma, \mathcal{D}_{\text{off}}) < \text{SEC}_{\text{RLHF}}(\Pi, T, \beta, \pi_{\text{samp}}).$$

In particular, $C_{\text{off}}^{(\pi)}$ is a measure of coverability for an arbitrary π, contributed by the offline dataset.

4.4 Main Results

We state the main sample complexity result of HPO below.

Theorem 1. Suppose Assumption 1 and 2 hold. For any $\beta > 0$ and $T \in [N]$, if we set $\alpha = c \cdot \frac{\beta}{(V_{\max}+R_{\max})e^{2R_{\max}}} \cdot \sqrt{\frac{\log(|\Pi|T\delta^{-1})\log(T)}{(T+\gamma)\cdot \text{SEC}_{\text{HybRLHF}}(\Pi,T,\beta,\pi_{\text{samp}};\gamma,\mathcal{D}_{\text{off}})}}$ for some absolute constant $c > 0$, then Algorithm 1 ensures that with probability at least $1-\delta$, we have

$$J_\beta(\pi_\beta^*) - J_\beta(\widehat{\pi}) \lesssim (V_{\max} + R_{\max})e^{2R_{\max}} \times \sqrt{\frac{(1+\gamma/T)\cdot \mathsf{SEC} \cdot \log(|\Pi|T\delta^{-1})\log(T)}{T}},$$

where $\mathsf{SEC} = \text{SEC}_{\text{HybRLHF}}(\Pi, T, \beta, \pi_{\text{samp}}; \gamma, \mathcal{D}_{\text{off}})$.

Note that $\gamma = 0$ represents the pure online setting, which is studied in XPO (Xie et al., 2024). Here, we can notice that the sample complexity is reduced because the SEC for hybrid RLHF becomes smaller than its pure online counterpart. That is, the size of the space that needs to be searched through online exploration becomes smaller because of the available offline dataset.

5 Breaking Lower Bounds through Hybrid Learning

In order to paint a clearer picture of why Hybrid RLHF is more sample-efficient than pure online and pure offline RLHF, we consider the special case of linear MDPs and study the suboptimality gaps of hybrid, online and offline RLHF. Our main goal is to show that the upper bounds for hybrid RLHF are smaller than the minimax lower bounds for pure online and pure offline RLHF.

We begin by formally defining the linear MDP setting.

Definition 5.1. (Linear MDP (Jin et al., 2020)) In a Linear MDP, the transition probability as well as the reward are linear functions of a known feature map $\phi(s,a) \in \mathbb{R}^d$, where $\|\phi\|_2 \leq 1$ and

$$P(s'|s,a) = \phi(s,a)^\top \mu(s') \quad r(s,a) = \phi(s,a)^\top \nu.$$

Here, $\mu(s')$ is an unknown feature map with $\|\sum_{s'} \mu(s')\|_2 \leq \sqrt{d}$, and $\nu \in \mathbb{R}^d$ is an unknown parameter with $\|\nu\|_2 \leq 1$.

5.1 Lower Bounds for Pure Offline and Online RLHF

Define $\Lambda_{\text{off}} = \frac{1}{N_{\text{off}}} \sum_{(\tau,\tilde{\tau})\in\mathcal{D}_{\text{off}}} (\phi(\tau) - \phi(\tilde{\tau}))(\phi(\tau) - \phi(\tilde{\tau}))^\top$ as the empirical feature covariance matrix of the offline preference dataset. Let $\nu^* = \frac{\mathbb{E}_{s\sim\rho,\tau\sim\pi_\beta^*(\cdot|s)}[\phi(s,\tau)]}{\|\mathbb{E}_{s\sim\rho,\tau\sim\pi_\beta^*(\cdot|s)}[\phi(s,\tau)]\|_2}$ denote a unit vector corresponding to the feature projection along the optimal policy π_β^*. We define the following quantity, for a given instance, i.e. a linear MDP $\mathcal{M} = \{\phi, \mu, \nu\}$ and offline dataset $\mathcal{D}_{\text{off}}$:

$$\text{C}^*(\mathcal{M}, \mathcal{D}_{\text{off}}) = \|\Lambda_{\text{off}}^{-\frac{1}{2}} \nu^*\|_2.$$

This measures the coverage of the offline dataset with respect to the optimal policy π_β^*. A Larger value indicates poorer coverage. Since for any arbitrary instance this quantity may be unbounded above, we consider all families of linear MDPs such that the concentrability is at most Δ, i.e.

$$\text{CB}(\Delta) = \{\mathcal{M}, \mathcal{D}_{\text{off}} | \text{C}^*(\mathcal{M}, \mathcal{D}_{\text{off}}) \leq \Delta\}.$$

Then by extending results from Li et al. (2022), there exists an instance in $\mathrm{CB}(\Delta)$ such that the suboptimality for any algorithm using ℓ_2 confidence sets is lower bounded by :

$$J_\beta(\pi^*_\beta) - J_\beta(\widehat{\pi}) \geq \Omega\left(\sqrt{\frac{d}{N_{\text{off}}}}\Delta\right).$$

Thus, by choosing $\Delta = \mathcal{O}(\sqrt{d})$ even in the restricted class $\mathrm{CB}(\mathcal{O}(\sqrt{\mathrm{d}}))$, the suboptimality lower bound is $\Omega\left(\sqrt{\frac{d^2}{N_{\text{off}}}}\right)$. We formally state this as a minimax lower bound for the pure offline case:

Theorem 2 (Sample Complexity Lower Bound for Offline RLHF)**.** For any algorithm $\mathcal{A}$, there exists an instance of offline RLHF problem and choice of β, such that if $\hat{\pi}^{\mathcal{A}}_n$ is the policy returned by $\mathcal{A}$ after any $n \geq d^2$ pairwise preference samples, then $J_\beta(\pi^*_\beta) - J_\beta(\hat{\pi}^{\mathcal{A}}_n) \geq \Omega\left(\sqrt{\frac{d^2}{n}}\right)$.

We now state a minimax lower bound for the pure online case:

Theorem 3 (Sample Complexity Lower Bound for Online RLHF)**.** For any algorithm $\mathcal{A}$, there exists an instance of online RLHF problem and choice of β, such that if $\hat{\pi}^{\mathcal{A}}_T$ is the policy returned by $\mathcal{A}$ after T rounds, then $J_\beta(\pi^*_\beta) - J_\beta(\hat{\pi}^{\mathcal{A}}_T) \geq \Omega\left(\sqrt{\frac{d^2}{T}}\right)$.

The proof of theorem 2 and theorem 3 respectively follows the existing lower bounds for online (Wagenmaker et al., 2022, Theorem 2) and offline (Li et al., 2022, Theorem 2) linear contextual bandits along with the reduction to idea argued in (Saha, 2021, Lemma 8). We have added the detailed proofs in appendix B.

5.2 Upper Bounds for Hybrid RLHF

Define $\tilde{\Lambda}_{\text{off}} = \Lambda_{\text{off}} + \frac{V^2_{\max}}{\gamma}\mathbf{I}$ as the empirical coverage. Let $\lambda^{(1)}_{\text{off}}, \ldots, \lambda^{(d)}_{\text{off}}$ be the eigenvalues of $\tilde{\Lambda}_{\text{off}}$. Let $\{v_1, v_2, \ldots, v_d\}$ denote the orthonormal eigenvectors corresponding to $\tilde{\Lambda}_{\text{off}}$. We adopt the convention that the eigenvalues corresponding to these eigenvectors are in increasing order. For a fixed threshold λ we can define:

$$d_{\text{hyb}} = \max_{i \in [d]} \left\{ \|\tilde{\Lambda}^{-\frac{1}{2}}_{\text{off}} v_i\|_2 \lesssim \Omega\left(\frac{1}{\sqrt{T}}\right) \right\}.$$

d_{hyb} measures the coverage of the offline dataset. A larger value of d_{hyb} indicates poor coverage. Note that the definition of d_{hyb} is also equivalent to $\max_{i \in [d]} \lambda^{(i)}_{\text{off}} \leq \Omega(1/T)$. We first state an upper bound on the sample complexity of HPO.

Theorem 4. By choosing $\gamma = \mathcal{O}(T)$, we then have

$$J_\beta(\pi^*_\beta) - J_\beta(\widehat{\pi}) \lesssim \tilde{\mathcal{O}}\left(R_{\max} e^{2R_{\max}} \sqrt{\frac{d \cdot d_{\text{hyb}}}{T}}\right).$$

Proof. We highlight the key steps of this proof. Using the elliptical potential lemma Lattimore and Szepesvári (2020), we can derive $\mathrm{SEC}_{\mathrm{HybRLHF}}(\Pi, T, \beta, \pi_{\text{samp}}; \gamma, \mathcal{D}_{\text{off}})$

$$\leq 2 \sum_{i \in [d]} \left(\log\left(1 + \frac{4T}{\gamma \lambda^{(i)}_{\text{off}}}\right)\right).$$

Choosing $\gamma = \mathcal{O}(T)$, then only terms with $\lambda^{(i)}_{\text{off}} \leq \Omega(\frac{1}{T})$ will remain in the summand, corresponding to the directions where the offline dataset has poor exploration. By defintion, d_{hyb} denote the number of indices in $[d]$ such that $\lambda^{(i)}_{\text{off}} \leq \Omega(\frac{1}{T})$. Noting that $\gamma \lambda^{(i)}_{\text{off}} \geq V^2_{\max} \quad \forall i \in [d]$, we get $\mathrm{SEC}_{\mathrm{HybRLHF}}(\Pi, T, \beta, \pi_{\text{samp}}; \gamma, \mathcal{D}_{\text{off}}) \leq \mathcal{O}\left(d_{\text{hyb}} \log\left(1 + \frac{4T}{V^2_{\max}}\right)\right)$. Hence, the SEC scales as the effective number of dimensions yet to be explored sufficiently. Plugging this into Theorem 1 gives the desired result. □

Note that d_{hyb} is always upper bounded by d. Thus for any arbitrary instance any arbitrary offline dataset, this upper bound is still $\tilde{\mathcal{O}}\left(\sqrt{\frac{d^2}{T}}\right)$. Furthermore, from the above analysis, it is clear that compared with pure offline and pure online RLHF, HPO has the following two advantages:

1. The upper bound of HPO in Theorem 4 beats the lower bounds of both pure online and offline RLHF in Theorem 2 and 3, as long as d_{hyb} is non-trivially smaller than d. That is, our proposed hybrid RLHF is provably better than both pure online and offline RLHF.
2. To achieve this upper bound, HPO does not require any assumptions in all-policy or single-policy concentrability coefficient, which is usually imposed in traditional offline RL (Zhan et al., 2022). To satisfy these kinds of assumptions, the behavior policy used to collect the offline dataset needs to cover all possible actions of the optimal policy, which can sometimes be stringent. However, HPO can effectively utilize the offline dataset even if the behavior policy covers no action of the optimal policy as d_{hyb} shrinks even through coverage in the offline datasets collected via by suboptimal policies.

6 Experiments

We evaluate the benefits of HPO on the linear contextual bandit setting considered in (Cen et al., 2024). We restate the salient features of their setup here. We consider a linear contextual bandit problem, where we set the prompt space as $\mathcal{X} = \mathbb{R}^2$ and the response space as $|\mathcal{Y}| = 500$ one-hot vectors. For each (x, y) pair, the ground truth reward is given by $r^\star(x, y) = \langle \phi(x, y), \theta^\star \rangle$, where $\theta^\star \in \mathbb{R}^{100}$ is randomly sampled from $\mathcal{U}([0, 1])$, and the feature vector $\phi(x, y)$ is the output of the hidden layer of a fixed two-layer MLP, with the input given by the concatenation of x and the one-hot encoding of y. The activation function is set to $\tanh$. The context vector x is drawn from standard normal distribution.

We focus on log-linear policy class $\pi_\theta(\cdot|x) = \text{softmax}(\langle \theta, \phi(\text{x}, \cdot) \rangle)$, and set $\pi_{\text{ref}} = \pi_{\theta_{\text{ref}}}$ with $\theta_{\text{ref}}(x, y)$ sampled i.i.d. from $\mathcal{U}([0, 1])$.

Offline Data Collection Policy: We collected response pairs from the reference policy π_{ref}.

Optimization Procedure: We use mini-batch samples of size 5 in every iteration. We approximately solve the optimization problems by performing 20 AdamW optimization steps with learning rate 0.01 and weight decay rate 0.01 in every iteration for the online setting and 1000 steps for the offline setting.

Choice of Hyperparameters: For Online VPO, we tried hyperparameters $\alpha = \{1.0, 10.0\}$ and for offline VPO we tried hperparameters $\alpha = \{\frac{0.1}{\sqrt{N_{\text{off}}}}, \frac{1.0}{\sqrt{N_{\text{off}}}}, \frac{10.0}{\sqrt{N_{\text{off}}}}\}$ and report the results for the best performing hyperparameter here. For HPO we try the same set of α's as Online VPO and set $\gamma = N_{\text{off}} = 500$.

We ask the following question: *Given an offline dataset where algorithms like DPO (Rafailov et al., 2023), Offline-VPO (Cen et al., 2024) fail to obtain a near optimal policy, can we still utilize information from the dataset to significantly reduce the number of online exploration samples needed to obtain a near optimal policy via HPO as compared to baselines such as Online DPO (Guo et al., 2024), Online VPO (Cen et al., 2024) (XPO (Xie et al., 2024))?*

We answer this question affirmatively. In particular, we note the following takeaways from our experiment:

1. In Figure **??** (left), the cumulative regret of HPO grows much slower compared to pure online learning baseline algorithms. The value plotted for any T with $T > N_{\text{off}}$ is $\sum_{i=500}^{T} \left(J_\beta(\pi^*_\beta) - J_\beta(\pi_i)\right)$. This ensures the online algorithms have seen online T samples, when we are talking about the cumulative regret at T, whereas the hybrid algorithm has seen N_{off} offline samples and $T - N_{\text{off}}$ offline samples. This ensures fairness while comparing the online and hybrid approaches.
2. In Figure **??** (right), we plot the suboptimality gaps as a function of total number of samples (online, offline or a mix of both). For the hybrid setting, we use an offline dataset of size $N_{\text{off}} = 500$. Note that the suboptimality gaps for offline algorithms run on this size of a dataset is quite poor compared to other (purely online) baselines. However, using the same

dataset, the suboptimality gaps for the policy produced by HPO is much better than all baselines.

7 Conclusion

We introduced Hybrid Preference Optimization (HPO), a novel approach that effectively integrates the strengths of both online and offline Reinforcement Learning from Human Feedback (RLHF) methods. By combining online exploration with existing offline preference data, HPO addresses the limitations inherent in each approach when used in isolation. Specifically, it relaxes the stringent concentrability conditions required by offline methods and enhances the sample efficiency of online exploration. Our theoretical analysis provides the first provably optimal bounds for hybrid RLHF with preference feedback, demonstrating that HPO achieves better sample complexity than either pure offline or pure online methods. By leveraging the wealth of existing offline data while simultaneously adapting to new information through online exploration, HPO reduces the need for extensive and costly real-time human feedback, making the process more scalable and less resource-intensive.

Acknowledgements

MF was supported in part by awards NSF TRIPODS II 2023166, CCF 2007036, CCF 2212261, CCF 2312775. SSD acknowledges the support of NSF IIS 2110170, NSF DMS 2134106, NSF CCF 2212261, NSF IIS 2143493, NSF IIS 2229881, Alfred P. Sloan Research Fellowship, and Amazon.

References

Alekh Agarwal, Nan Jiang, and Sham M Kakade. Reinforcement learning: Theory and algorithms. *Preprint*, 2019.

Philip Amortila, Dylan J Foster, Nan Jiang, Ayush Sekhari, and Tengyang Xie. Harnessing density ratios for online reinforcement learning. *arXiv preprint arXiv:2401.09681*, 2024.

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

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Viktor Bengs, Aadirupa Saha, and Eyke Hüllermeier. Stochastic contextual dueling bandits under linear stochastic transitivity models. In *International Conference on Machine Learning*, pages 1764–1786. PMLR, 2022.

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

Shicong Cen, Jincheng Mei, Katayoon Goshvadi, Hanjun Dai, Tong Yang, Sherry Yang, Dale Schuurmans, Yuejie Chi, and Bo Dai. Value-incentivized preference optimization: A unified approach to online and offline rlhf. *arXiv preprint arXiv:2405.19320*, 2024.

Jonathan D Chang, Wenhao Shan, Owen Oertell, Kianté Brantley, Dipendra Misra, Jason D Lee, and Wen Sun. Dataset reset policy optimization for rlhf. *arXiv preprint arXiv:2404.08495*, 2024.

Xiaoyu Chen, Han Zhong, Zhuoran Yang, Zhaoran Wang, and Liwei Wang. Human-in-the-loop: Provably efficient preference-based reinforcement learning with general function approximation. In *International Conference on Machine Learning*, pages 3773–3793. PMLR, 2022.

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

Yihan Du, Anna Winnicki, Gal Dalal, Shie Mannor, and R Srikant. Exploration-driven policy optimization in RLHF: Theoretical insights on efficient data utilization. *arXiv preprint arXiv:2402.10342*, 2024.

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

Dylan J Foster and Alexander Rakhlin. Foundations of reinforcement learning and interactive decision making. *arXiv preprint arXiv:2312.16730*, 2023.

Zhaolin Gao, Jonathan D Chang, Wenhao Zhan, Owen Oertell, Gokul Swamy, Kianté Brantley, Thorsten Joachims, J Andrew Bagnell, Jason D Lee, and Wen Sun. REBEL: Reinforcement learning via regressing relative rewards. *arXiv preprint arXiv:2404.16767*, 2024.

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. Direct language model alignment from online AI feedback. *arXiv preprint arXiv:2402.04792*, 2024.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143, 2020.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on robot learning*, pages 651–673. PMLR, 2018.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Gene Li, Cong Ma, and Nati Srebro. Pessimism for offline linear contextual bandits using ℓ_p confidence sets. *Advances in Neural Information Processing Systems*, 35:20974–20987, 2022.

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

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*, 2024.

Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Andrea Michi, et al. Nash learning from human feedback. *arXiv preprint arXiv:2312.00886*, 2023.

Sahand Negahban, Sewoong Oh, and Devavrat Shah. Iterative ranking from pair-wise comparisons. *Advances in neural information processing systems*, 25, 2012.

Andi Nika, Debmalya Mandal, Parameswaran Kamalaruban, Georgios Tzannetos, Goran Radanović, and Adish Singla. Reward model learning vs. direct policy optimization: A comparative analysis of learning from human preferences. *arXiv preprint arXiv:2403.01857*, 2024.

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

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

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to $Q^\star$: Your language model is secretly a Q-function. *arXiv preprint arXiv:2404.12358*, 2024.

Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacroce, Ahmed Awadallah, and Tengyang Xie. Direct Nash Optimization: Teaching language models to self-improve with general preferences. *arXiv preprint arXiv:2404.03715*, 2024.

Aadirupa Saha. Optimal algorithms for stochastic contextual preference bandits. *Advances in Neural Information Processing Systems*, 34:30050–30062, 2021.

Ruizhe Shi, Runlong Zhou, and Simon S Du. The crucial role of samplers in online direct preference optimization. *arXiv preprint arXiv:2409.19605*, 2024.

Yuda Song, Yifei Zhou, Ayush Sekhari, J Andrew Bagnell, Akshay Krishnamurthy, and Wen Sun. Hybrid rl: Using both offline and online data can make rl efficient. *arXiv preprint arXiv:2210.06718*, 2022.

Yuda Song, Gokul Swamy, Aarti Singh, J Andrew Bagnell, and Wen Sun. Understanding preference fine-tuning through the lens of coverage. *arXiv preprint arXiv:2406.01462*, 2024.

Kevin Tan, Wei Fan, and Yuting Wei. Hybrid reinforcement learning breaks sample size barriers in linear mdps. *arXiv preprint arXiv:2408.04526*, 2024.

Andrew J Wagenmaker, Yifang Chen, Max Simchowitz, Simon Du, and Kevin Jamieson. Reward-free rl is no harder than reward-aware rl in linear markov decision processes. In *International Conference on Machine Learning*, pages 22430–22456. PMLR, 2022.

Yuanhao Wang, Qinghua Liu, and Chi Jin. Is RLHF more difficult than standard RL? *arXiv preprint arXiv:2306.14111*, 2023.

Runzhe Wu and Wen Sun. Making RL with preference-based feedback efficient via randomization. *arXiv preprint arXiv:2310.14554*, 2023.

Tianhao Wu, Banghua Zhu, Ruoyu Zhang, Zhaojin Wen, Kannan Ramchandran, and Jiantao Jiao. Pairwise proximal policy optimization: Large language models alignment via comparative rl. 2024.

Tengyang Xie, Nan Jiang, Huan Wang, Caiming Xiong, and Yu Bai. Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. *Advances in neural information processing systems*, 34:27395–27407, 2021.

Tengyang Xie, Dylan J Foster, Yu Bai, Nan Jiang, and Sham M Kakade. The role of coverage in online reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Tengyang Xie, Dylan J Foster, Akshay Krishnamurthy, Corby Rosset, Ahmed Awadallah, and Alexander Rakhlin. Exploratory preference optimization: Harnessing implicit q*-approximation for sample-efficient rlhf. *arXiv preprint arXiv:2405.21046*, 2024.

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

Jing Xu, Andrew Lee, Sainbayar Sukhbaatar, and Jason Weston. Some things are more cringe than others: Preference optimization with the pairwise cringe loss. *arXiv preprint arXiv:2312.16682*, 2023.

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

Wenhao Zhan, Baihe Huang, Audrey Huang, Nan Jiang, and Jason Lee. Offline reinforcement learning with realizability and single-policy concentrability. In *Conference on Learning Theory*, pages 2730–2775. PMLR, 2022.

Wenhao Zhan, Masatoshi Uehara, Nathan Kallus, Jason D Lee, and Wen Sun. Provable offline preference-based reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2023.

Shenao Zhang, Donghan Yu, Hiteshi Sharma, Ziyi Yang, Shuohang Wang, Hany Hassan, and Zhaoran Wang. Self-exploring language models: Active preference elicitation for online alignment, 2024.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*, 2023.

Han Zhong, Wei Xiong, Sirui Zheng, Liwei Wang, Zhaoran Wang, Zhuoran Yang, and Tong Zhang. GEC: A unified framework for interactive decision making in MDP, POMDP, and beyond. *arXiv preprint arXiv:2211.01962*, 2022.

Banghua Zhu, Michael Jordan, and Jiantao Jiao. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In *International Conference on Machine Learning*, pages 43037–43067. PMLR, 2023.

Banghua Zhu, Michael I Jordan, and Jiantao Jiao. Iterative data smoothing: Mitigating reward overfitting and overoptimization in rlhf. *arXiv preprint arXiv:2401.16335*, 2024.

A Upper Bound Proofs

The proof structure for Theorem 1 is similar to Theorem 3.1 (Xie et al., 2024). However due to the hybrid nature of our algorithm, we are able to derive concentration lemmas with a faster convergence rate. We highlight the main differences in our analysis, which lead to the reduced sample complexity.

A.1 Concentration Lemmas

Let $\mathbb{P}_{\text{off}}$ denote the nominal distribution of trajectory pairs in the offline dataset $\mathcal{D}_{\text{off}}$. Define $\boldsymbol{\mu}_{\text{hyb}}^{(t)} = \frac{1}{t-1+\gamma}(\sum_{i=1}^{t-1} \pi^{(i)} \otimes \tilde{\pi}^{(i)} + \gamma \mathbb{P}_{\text{off}})$.

Define

$$f_\pi(\tau, \tilde{\tau}) = \beta \log \frac{\pi(\tau)}{\pi_{\text{ref}}(\tau)} - \beta \log \frac{\pi(\tilde{\tau})}{\pi_{\text{ref}}(\tilde{\tau})}.$$

Lemma 1 (Concentration for Algorithm 1)**.** Suppose that Assumptions 1,2 hold. Then Algorithm 1 guarantees that with probability at least $1-\delta$, for all steps $t \in [T]$,

$$\alpha \cdot \mathbb{E}_{s_1 \sim \rho, \tau \sim \tilde{\pi}^{(t-1)}}[\log(\pi^{(t)}(\tau)) - \log(\pi_\beta^\star(\tau))] + \kappa \cdot \left[\mathbb{E}_{s_1 \sim \rho, (\tau,\tilde{\tau}) \sim \boldsymbol{\mu}_{\text{hyb}}^{(t)} | s_1}[f_{\pi^{(t)}}(\tau, \tilde{\tau}) - f_{\pi_\beta^\star}(\tau, \tilde{\tau})]^2\right]$$
$$\leq \frac{2\log(2|\Pi|T\delta^{-1})}{\gamma + t - 1} + \frac{\alpha}{\beta} V_{\max} \sqrt{\frac{2^4 \log(2|\Pi|T\delta^{-1})}{\gamma + t - 1}},$$

for $\kappa = (8(R_{\max} + V_{\max})e^{2R_{\max}})^{-2}$.

Proof. Fix $t \in \{2, \ldots, T+1\}$.

We wish to decompose the objective in HPO, Eq equation 5 into 2 terms as defined below: Define the following quantities:

$$\pi^{(t)} = \arg\min_{\pi \in \Pi} \left[\widehat{L}^{(t)}(\pi) + \widehat{B}^{(t)}(\pi)\right]. \tag{10}$$

$$\widehat{L}^{(t)}(\pi) = \sum_{(\tau^+, \tau^-) \in \mathcal{D}_{\text{hyb}}^{(t)}} \log \left[\sigma(\beta \log \frac{\pi(\tau^+)}{\pi_{\text{ref}}(\tau^+)} - \beta \log \frac{\pi(\tau^-)}{\pi_{\text{ref}}(\tau^-)})\right].$$

$$\widehat{B}^{(t)}(\pi) = \alpha \sum_{\tau \in \mathcal{D}_{\text{opt}}^{(t)}} \log \pi(\tau).$$

Now we provide concentration lemmas for these individual terms before combining them together.

Lemma 2. For any fixed $t \geq 1$, for all $\pi \in \Pi$ with probability at least $1-\delta$, we have:

$$\kappa \cdot \left[\mathbb{E}_{s_1 \sim \rho, (\tau,\tilde{\tau}) \sim \boldsymbol{\mu}_{\text{hyb}}^{(t)} | s_1}[f_{\pi^{(t)}}(\tau, \tilde{\tau}) - f_{\pi_\beta^\star}(\tau, \tilde{\tau})]^2\right] \leq \widehat{L}^{(t)}(\pi) - \widehat{L}^{(t)}(\pi_\beta^\star) + \log(2|\Pi|T\delta^{-1})$$

Proof. Use the definition of $\boldsymbol{\mu}_{\text{hyb}}^{(t)}$ to see the dataset $\mathcal{D}_{\text{hyb}}^{(t)}$ is a sequence adapted to the filtration $\mathcal{F}^{(\gamma+t)} = \sigma(\mathcal{D}_{\text{off}}, \ldots, \mathcal{D}_{\text{off}}, (\tau^{(1)}, \tilde{\tau}^1), \ldots, (\tau^{(t-1)}, \tilde{\tau}^{(t-1)}))$. Then apply the Martinagle Chernoff Theorem to get a high probability bound. Combine it with Lemma C.8 (Xie et al., 2024) to get the above result. □

Lemma 3. For any fixed $t \geq 1$, for all $\pi \in \Pi$ with probability at least $1-\delta$, we have:

$$\alpha \cdot (t + \gamma - 1) \cdot \mathbb{E}_{s_1 \sim \rho, \tau \sim \tilde{\pi}^{(t-1)}}[\log(\pi^{(t)}(\tau)) - \log(\pi_\beta^\star(\tau))] \leq \widehat{B}^{(t)}(\pi) - \widehat{B}^{(t)}(\pi_\beta^\star) + \frac{\alpha}{\beta} V_{\max} \sqrt{2^4 \log(2|\Pi|\delta^{-1})}$$

Proof. Apply Azuma Hoeffding with its sample average over $t + \gamma$ terms. □

Combining Lemma 2 and Lemma 3, taking an union bound over all time steps $t \in [T]$, and utilizing the definition of $\pi^{(t)}$ in Eq. equation 10 to note $\widehat{B}^{(t)}(\pi) + \widehat{L}^{(t)}(\pi) \leq \widehat{B}^{(t)}(\pi_\beta^\star) + \widehat{L}^{(t)}(\pi_\beta^\star)$, get we get the stated result. □

A.2 Proof of Theorem 1

We use the standard steps for regret decomposition, and similar proof idea in (Xie et al., 2024). The main difference in our analysis is the definition of the term $\mathcal{I}^{(t)}$, which allows us to use the faster convergence concentration lemma in Lemma 1, and also use our definition of the SEC in the hybrid RLHF case, $\mathrm{SEC}_{\mathrm{HybRLHF}}(\Pi, T, \beta, \pi_{\mathrm{samp}}; \gamma, \mathcal{D}_{\mathrm{off}})$, which is smaller than the SEC in the online case.

Theorem 1. Suppose Assumption 1 and 2 hold. For any $\beta > 0$ and $T \in [N]$, if we set $\alpha = c \cdot \frac{\beta}{(V_{\max}+R_{\max})e^{2R_{\max}}} \cdot \sqrt{\frac{\log(|\Pi|T\delta^{-1})\log(T)}{(T+\gamma)\cdot \mathrm{SEC}_{\mathrm{HybRLHF}}(\Pi,T,\beta,\pi_{\mathrm{samp}};\gamma,\mathcal{D}_{\mathrm{off}})}}$ for some absolute constant $c > 0$, then Algorithm 1 ensures that with probability at least $1-\delta$, we have

$$J_\beta(\pi^*_\beta) - J_\beta(\widehat{\pi}) \lesssim (V_{\max} + R_{\max})e^{2R_{\max}} \times \sqrt{\frac{(1+\gamma/T)\cdot \mathsf{SEC} \cdot \log(|\Pi|T\delta^{-1})\log(T)}{T}},$$

where $\mathsf{SEC} = \mathrm{SEC}_{\mathrm{HybRLHF}}(\Pi, T, \beta, \pi_{\mathrm{samp}}; \gamma, \mathcal{D}_{\mathrm{off}})$.

Proof. Using the regret decomposition techniques for KL-Regularized MDPs from (Xie et al., 2024), we derive:

$$\begin{aligned}
&J_\beta(\pi^*_\beta) - J_\beta(\pi^{(t)}) \\
&\le \frac{6V_{\max}}{T} + \frac{1}{T}\sum_{t=2}^{T} \mathbb{E}_{\tau\sim\tilde{\pi}^{(t-1)}}\left[\beta\log(\pi^{(t)}(\tau)) - \beta\log(\pi^\star_\beta(\tau))\right] \\
&+ \frac{1}{T}\sum_{t=2}^{T} \mathbb{E}_{s_1\sim\rho,\tau\sim\pi^{(t)},\tilde{\tau}\sim\tilde{\pi}^{(t)}}\left[\beta\log\frac{\pi^{(t)}(\tau)}{\pi_{\mathrm{ref}}(\tau)} - r(\tau) - \beta\log\frac{\pi^{(t)}(\tilde{\tau})}{\pi_{\mathrm{ref}}(\tilde{\tau})} + r(\tilde{\tau})\right]
\end{aligned}$$

Recalling the definition of $\boldsymbol{\mu}^{(t)}_{\mathrm{hyb}} = \frac{1}{t-1+\gamma}(\sum_{i=1}^{t-1}\pi^{(i)}\otimes\tilde{\pi}^{(i)} + \gamma\mathbb{P}_{\mathrm{off}})$, we define:

$$\mathcal{I}^{(t)} = \frac{\left(\mathbb{E}_{s_1\sim\rho,\tau\sim\pi^{(t)},\tilde{\tau}\sim\tilde{\pi}^{(t)}}\left[\beta\log\frac{\pi^{(t)}(\tau)}{\pi_{\mathrm{ref}}(\tau)} - r(\tau) - \beta\log\frac{\pi^{(t)}(\tilde{\tau})}{\pi_{\mathrm{ref}}(\tilde{\tau})} + r(\tilde{\tau})\right]\right)^2}{V^2_{\max}\vee(t-1+\gamma)\cdot\mathbb{E}_{s_1\sim\rho,\tau,\tilde{\tau}\sim\boldsymbol{\mu}^{(t)}_{\mathrm{hyb}}|s_1}\left[\left(\beta\log\frac{\pi^{(t)}(\tau)}{\pi_{\mathrm{ref}}(\tau)} - r(\tau) - \beta\log\frac{\pi^{(t)}(\tilde{\tau})}{\pi_{\mathrm{ref}}(\tilde{\tau})} + r(\tilde{\tau})\right)^2\right]}$$

Using AM-GM inequality we can bound:

$$\begin{aligned}
&\mathbb{E}_{s_1\sim\rho,\tau\sim\pi^{(t)},\tilde{\tau}\sim\tilde{\pi}^{(t)}}\left[\beta\log\frac{\pi^{(t)}(\tau)}{\pi_{\mathrm{ref}}(\tau)} - r(\tau) - \beta\log\frac{\pi^{(t)}(\tilde{\tau})}{\pi_{\mathrm{ref}}(\tilde{\tau})} + r(\tilde{\tau})\right] \\
&\le \frac{\mathcal{I}^{(t)}}{2\eta} + \frac{\eta}{2}\cdot\left(V^2_{\max}\vee(t-1+\gamma)\cdot\mathbb{E}_{s_1\sim\rho,\tau,\tilde{\tau}\sim\boldsymbol{\mu}^{(t)}_{\mathrm{hyb}}}\left[\left(\beta\log\frac{\pi^{(t)}(\tau)}{\pi_{\mathrm{ref}}(\tau)|s_1} - r(\tau) - \beta\log\frac{\pi^{(t)}(\tilde{\tau})}{\pi_{\mathrm{ref}}(\tilde{\tau})} + r(\tilde{\tau})\right)^2\right]\right) \\
&\le \frac{\mathcal{I}^{(t)}}{2\eta} + \frac{\eta}{2}\cdot\left(V^2_{\max} + (t-1+\gamma)\cdot\mathbb{E}_{s_1\sim\rho,\tau,\tilde{\tau}\sim\boldsymbol{\mu}^{(t)}_{\mathrm{hyb}}|s_1}\left[\left(\beta\log\frac{\pi^{(t)}(\tau)}{\pi_{\mathrm{ref}}(\tau)} - r(\tau) - \beta\log\frac{\pi^{(t)}(\tilde{\tau})}{\pi_{\mathrm{ref}}(\tilde{\tau})} + r(\tilde{\tau})\right)^2\right]\right)
\end{aligned}$$

Recall the definition of $\mathrm{SEC}_{\mathrm{HybRLHF}}(\Pi, T, \beta, \pi_{\mathrm{samp}}; \gamma, \mathcal{D}_{\mathrm{off}})$, and note that $\sum_{t=1}^{T}\mathcal{I}^{(t)} \le \mathrm{SEC}_{\mathrm{HybRLHF}}(\Pi, T, \beta, \pi_{\mathrm{samp}}; \gamma, \mathcal{D}_{\mathrm{off}})$.

Now we can write:

$$\begin{aligned}
&J_\beta(\pi^*_\beta) - J_\beta(\pi^{(t)}) \\
&\le \frac{6V_{\max}}{T} + \frac{\mathrm{SEC}_{\mathrm{HybRLHF}}(\Pi, T, \beta, \pi_{\mathrm{samp}}; \gamma, \mathcal{D}_{\mathrm{off}})}{2\eta T} + \frac{\eta}{2}V^2_{\max} + \frac{1}{T}\sum_{t=2}^{T}\left(\mathbb{E}_{\tau\sim\tilde{\pi}^{(t-1)}}\left[\beta\log(\pi^{(t)}(\tau)) - \beta\log(\pi^\star_\beta(\tau))\right]\right. \\
&\left.+ \frac{\eta}{2}\cdot(t-1+\gamma)\cdot\mathbb{E}_{s_1\sim\rho,\tau,\tilde{\tau}\sim\boldsymbol{\mu}^{(t)}_{\mathrm{hyb}}|s_1}\left[\left(\beta\log\frac{\pi^{(t)}(\tau)}{\pi_{\mathrm{ref}}(\tau)} - r(\tau) - \beta\log\frac{\pi^{(t)}(\tilde{\tau})}{\pi_{\mathrm{ref}}(\tilde{\tau})} + r(\tilde{\tau})\right)^2\right]\right)
\end{aligned}$$

For a fixed $t \in \{2, \ldots, T+1\}$, we consider the term:

$$\mathbb{E}_{\tau \sim \tilde{\pi}^{(t-1)}}\left[\beta \log(\pi^{(t)}(\tau)) - \beta \log(\pi^{\star}_{\beta}(\tau))\right] + \frac{\eta}{2} \cdot (t-1+\gamma) \cdot \mathbb{E}_{s_1 \sim \rho, \tau, \tilde{\tau} \sim \boldsymbol{\mu}^{(t)}_{\mathrm{hyb}} | s_1}\left[\left(\beta \log \frac{\pi^{(t)}(\tau)}{\pi_{\mathrm{ref}}(\tau)} - r(\tau) - \beta \log \frac{\pi^{(t)}(\tilde{\tau})}{\pi_{\mathrm{ref}}(\tilde{\tau})} + r(\tilde{\tau})\right)^2\right]$$

Setting $\eta = \frac{\beta\kappa}{\alpha(T+\gamma)}$, and invoking Lemma 1, we get:

$$\begin{aligned}
&J_{\beta}(\pi^{*}_{\beta}) - J_{\beta}(\pi^{(t)}) \\
&\lesssim \frac{V_{\max}}{T} + \frac{\alpha(T+\gamma)\mathrm{SEC}_{\mathrm{HybRLHF}}(\Pi, T, \beta, \pi_{\mathrm{samp}}; \gamma, \mathcal{D}_{\mathrm{off}})}{\beta\kappa T} + \frac{\beta\kappa}{\alpha(T+\gamma)} V^2_{\max} + \\
&\sum_{t=2}^{T} \frac{1}{T}\left(\frac{\beta}{\alpha}\frac{\log(|\Pi|T\delta^{-1})}{\gamma+t-1} + V_{\max}\sqrt{\frac{\log(|\Pi|T\delta^{-1})}{\gamma+t-1}}\right) \\
&\lesssim \frac{V_{\max}}{T} + \frac{\alpha(T+\gamma)\mathrm{SEC}_{\mathrm{HybRLHF}}(\Pi, T, \beta, \pi_{\mathrm{samp}}; \gamma, \mathcal{D}_{\mathrm{off}})}{\beta\kappa T} + \frac{\beta\kappa}{\alpha(T+\gamma)} V^2_{\max} + \\
&\frac{\beta}{\alpha}\frac{\log(|\Pi|T\delta^{-1})\log(T)}{T} + V_{\max}\sqrt{\frac{\log(|\Pi|T\delta^{-1})}{T}}
\end{aligned}$$

Choosing

$$\alpha \propto \frac{\beta}{(V_{\max} + R_{\max})e^{2R_{\max}}} \cdot \sqrt{\frac{\log(|\Pi|T\delta^{-1})\log(T)}{(T+\gamma) \cdot \mathrm{SEC}_{\mathrm{HybRLHF}}(\Pi, T, \beta, \pi_{\mathrm{samp}}; \gamma, \mathcal{D}_{\mathrm{off}})}}$$

and noting $\kappa < V^{-2}_{\max}$, we get:

$$\begin{aligned}
&J_{\beta}(\pi^{*}_{\beta}) - J_{\beta}(\pi^{(t)}) \\
&\leq \kappa^{-1}\sqrt{\frac{(1+\gamma/T) \cdot \mathrm{SEC}_{\mathrm{HybRLHF}}(\Pi, T, \beta, \pi_{\mathrm{samp}}; \gamma, \mathcal{D}_{\mathrm{off}}) \cdot \log(|\Pi|T\delta^{-1})\log(T)}{T}} + V_{\max}\sqrt{\frac{\log(|\Pi|T\delta^{-1})}{T}} \\
&\leq \mathcal{O}((V_{\max} + R_{\max})e^{2R_{\max}}) \cdot \sqrt{\frac{(1+\gamma/T) \cdot \mathrm{SEC}_{\mathrm{HybRLHF}}(\Pi, T, \beta, \pi_{\mathrm{samp}}; \gamma, \mathcal{D}_{\mathrm{off}}) \cdot \log(|\Pi|T\delta^{-1})\log(T)}{T}}.
\end{aligned}$$

□

B Lower Bound Proofs

In this section, we will analyze the proofs of theorem 3 and theorem 2. Our proofs rely on a key observation that the RLHF problem, as described in section 3 can essentially be seen as a BTL-based dueling bandit (DB) setting (Negahban et al., 2012; Bengs et al., 2022), both for the online and offline problem. For completeness, we first describe the preference model below:

BTL-based Pairwise Preference (Dueling) Model: Consider a decision space $\mathcal{D}$. Each action/item $\mathbf{a} \in \mathcal{D}$ is associated to a linear score/reward $s(\mathbf{a}) \in \mathbb{R}$. The probability of action $\mathbf{a}$ preferred over action $\mathbf{b}$ is given by:

$$P(\mathbf{a} \succ \mathbf{b}) = \big(\sigma\big(s(\mathbf{a}) - s(\mathbf{b})\big)\big),$$

where $\sigma(\cdot)$ being the sigmoid transformation given by $\sigma(x) = \frac{1}{1+e^{-x}}$ for any $x \in \mathbb{R}$. Note the above preference model exactly boils down to the BTL model-based preferences described in eq. (1) in section 3. We will denote this preference model as BTL-DB.

This essentially establishes the connection between the RLHF feedback model and BTL-based preference model BTL-DB (note that the trajectories (τ) lie in the action space $\mathcal{D}$ for the DB problem). Further note that the reward/ score function $s(\cdot)$ essentially corresponds to the $J(\cdot)$ function of eq. (2) (for $\beta = 0$). We next establish a connection between two well-studied feedback models in learning theory: *linear score BTL-DB* feedback and linear bandits (`linB`) feedback. We first describe the two feedback models below:

Linear score based BTL-DB (`linDB`) Feedback model. In addition to the modeling assumptions described above for BTL-DB, we further assume $\mathcal{D} \subset \mathbb{R}^d$ and the underlying score/reward function (s) is linear, i.e. $s(\mathbf{a}) = \langle \mathbf{a}, \mathbf{w} \rangle$, where $\mathbf{w} \in \mathbb{R}^d$ is an unknown direction in $\mathbb{R}^d$. We will also denote by $\mathbf{a}^* := \arg\max_{\mathbf{a} \in \mathcal{D}} s(\mathbf{a})$ the action with the highest score. The feedback model outputs a 1-bit binary feedback $o \sim \text{Ber}(P(\mathbf{a} \succ \mathbf{b}))$ upon receiving any pair of actions $(\mathbf{a}, \mathbf{b}) \in \mathcal{D} \times \mathcal{D}$ – we will use the abbreviation `linDB` for this feedback model.

Linear Bandits (`linB`) Feedback Model. Similar to the `linDB` setting above, we again assume a decision space $\mathcal{D} \subset \mathbb{R}^d$ with each arm $\mathbf{a} \in \mathcal{D}$ being associated to an underlying linear score/reward function $s(\mathbf{a}) = \langle \mathbf{a}, \mathbf{w} \rangle$, $\mathbf{w} \in \mathbb{R}^d$ is unknown. As oppose to the preference feedback observed in the `linDB` case, in this feedback model one gets to observe a noisy feedback of the true reward of any queried arm $\mathbf{a} \in \mathcal{D}$ given by $r(\mathbf{a}) = s(\mathbf{a}) + \eta$, where η is usually a 0 mean noise. For example, $\eta \sim \text{Gaussian}(0, 1)$ etc. We will use the abbreviation `linB` for this feedback model.

B.1 Simulating linDB feedback with linB Feedback.

Based on the reduction idea of (Saha, 2021, Lemma 8), we know that one can always simulate 1 unit of `linDB` pairwise preference feedback from 2 units of `linB` reward/score feedback. Formally the claim is given by:

Lemma 4. Given any two actions $\mathbf{a}$ and $\mathbf{b} \in \mathcal{D}$ if $r(\mathbf{a}) = s(\mathbf{a}) + \eta_1$ and $r(\mathbf{b}) = s(\mathbf{b}) + \eta_2$ are two (noisy) `linB` feedback, s.t. $\eta_1, \eta_2 \overset{\text{iid}}{\sim}$ Gumbel(0,1) are iid draws from Gumbel$(0, 1)$ distribution, then the binary outcome $o = \mathbf{1}(r(\mathbf{a}) > r(\mathbf{b})) \sim P(\mathbf{a} \succ \mathbf{b})$ follows `linDB` feedback.

Interested readers are encouraged to go over the proof of Lemma 9 of Saha (2021) to see the proof of lemma 4 above. Given the above result, we are now ready to prove the lower bound results of theorem 3 and theorem 2 as discussed in the following two subsections. We would first define the online and offline versions of the best-arm identification problem (BAI) with `linDB` feedback.

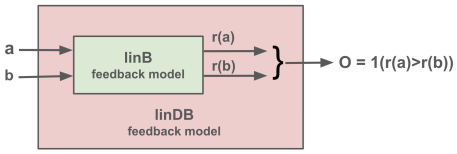

Figure 1: Simulating `linDB` feedback with `linB` feedback

B.2 Proof of Theorem 3

The *online version of BAI problem with `linDB` feedback* is an active sequential decision-making process where at each round t the learner (i.e. the algorithm) is supposed to play a pair of actions $(\mathbf{a}_t, \mathbf{b}_t) \in \mathcal{D} \times \mathcal{D}$, upon which it gets to see a binary preference feedback $o_t \sim P(\mathbf{a}_t \succ \mathbf{b}_t)$ is the preference feedback of the pair $(\mathbf{a}_t, \mathbf{b}_t)$ generated according to the `linDB` feedback model. Given a horizon of T rounds, the objective of the learner is to output an arm $\mathbf{a}_T \in \mathcal{D}$ such that $\mathbb{E}[r(\mathbf{a}_T)] = s(\mathbf{a}_t)$ is maximized or the difference $s(\mathbf{a}^*) - s(\mathbf{a}_T)$ is minimized. Let us denote this problem as *Online-BAI-*`linDB` problem.

Similarly, the *online version of BAI problem with `linB` feedback* is an active sequential decision-making process where at each round t the learner (i.e. the algorithm) is supposed to play an action $\mathbf{a}_t \in \mathcal{D}$, upon which it gets to see a real-valued reward/score feedback r_t generated according to the `linB` feedback model. Given a horizon of T rounds, the objective of the learner is to output an arm $\mathbf{a}_T \in \mathcal{D}$ such that $\mathbb{E}[r(\mathbf{a}_T)] = s(\mathbf{a}_t)$ is maximized or the difference $s(\mathbf{a}^*) - s(\mathbf{a}_T)$ is minimized. Let us denote this problem as *Online-BAI-*`linB` problem.

Key Idea: Reducing *Online-BAI-*`linB` to *Online-BAI-*`linDB`. Owing to lemma 4, it is easy to see that one could reduce the *Online-BAI-*`linB` problem to *Online-BAI-*`linDB` problem, i.e. given any algorithm $\mathcal{A}^{\texttt{linDB}}$ for the latter problem, one can use it to solve the former. We described the idea in algorithm 2 below to construct $\mathcal{A}^{\texttt{linB}}$—an *Online-BAI-*`linB` algorithm using $\mathcal{A}^{\texttt{linDB}}$.

Algorithm 2 Simulating $\mathcal{A}^{\texttt{linB}}$ using $\mathcal{A}^{\texttt{linDB}}$

Input: Time horizon T.
for $t = 1, 2, \ldots \lceil \frac{T}{2} \rceil$ **do**
  Receive: $(\mathbf{a}_t, \mathbf{b}_t) \leftarrow$ duel played by $\mathcal{A}^{\texttt{linDB}}$ at time t.
  Play $\mathbf{a}_t$ at round $(2t-1)$. Receive $r(\mathbf{a}_t)$.
  Play $\mathbf{b}_t$ at round $2t$. Receive $r(\mathbf{b}_t)$.
  Feedback: $o_t = \mathbf{1}(r(\mathbf{a}_t) > r(\mathbf{b}_t))$ to $\mathcal{A}^{\texttt{linDB}}$.
end for

But since a single round of *Online-BAI-*`linDB` corresponds to two rounds of *Online-BAI-*`linB` a valid BAI lower bound for the latter would immediately imply a BAI lower bound for the former problem. The result of theorem 3 now follows from the known existing lower bound for the *Online-BAI-*`linB` problem from (Wagenmaker et al., 2022, Theorem 2).

B.3 Proof of Theorem 2

The *offline version of BAI problem with* `linDB` *feedback* is a batch decision-making process where the learner is provided with an offline dataset $\mathcal{D}_{\text{off}} = \{(\mathbf{a}_i, \mathbf{b}_i, o_i)\}_{i=1}^n$, where for each triplet $(\mathbf{a}_i, \mathbf{b}_i, o_i)$, $(\mathbf{a}_i, \mathbf{b}_i) \in \mathcal{D} \times \mathcal{D}$ denotes a pair of action and $o_i \sim P(\mathbf{a}_i \succ \mathbf{b}_i)$ is the preference feedback of the pair $(\mathbf{a}_i, \mathbf{b}_i)$ generated according to the `linDB` feedback model. The objective of the learner is to output an arm $\mathbf{a}_n \in \mathcal{D}$ using the n samples of $\mathcal{D}_{\text{off}}$ such that $\mathbb{E}[r(\mathbf{a}_n)] = s(\mathbf{a}_t)$ is maximized or the difference $s(\mathbf{a}^*) - s(\mathbf{a}_n)$ is minimized. Let us denote this problem as *Offline-BAI-*`linDB` problem.

In the same spirit, the *offline version of BAI problem with* `linDB` *feedback* is a batch decision-making process where the learner is provided with an offline dataset $\mathcal{D}_{\text{off}} = \{(\mathbf{a}_i, r_i)\}_{i=1}^n$, where $\mathbf{a}_i \in \mathcal{D}$ is an arbitrary action in $\mathcal{D}$ and r_i is a noisy sample of the score/reward of arm i drawn according to the `linB` feedback model. The objective of the learner is to output an arm $\mathbf{a}_n \in \mathcal{D}$ using the n samples of $\mathcal{D}_{\text{off}}$ such that $\mathbb{E}[r(\mathbf{a}_n)] = s(\mathbf{a}_t)$ is maximized or the difference $s(\mathbf{a}^*) - s(\mathbf{a}_n)$ is minimized. Let us denote this problem as *Offline-BAI-*`linB` problem.

Reducing *Offline-BAI-*`linB` to *Offline-BAI-*`linDB`. Owing to lemma 4 and following a similar approach described in algorithm 2 above it is easy to see that one can obtain an algorithm for the *Offline-BAI-*`linB` problem using an algorithm for the *Offline-BAI-*`linDB` algorithm.

Following the same argument used for proving theorem 3 in appendix B.2 above, the result of theorem 2 now follows from the known existing lower bound for the *Offline-BAI-*`linB` problem from (Li et al., 2022, Theorem 2).