ADVERSARIAL POLICY OPTIMIZATION FOR OFFLINE PREFERENCE-BASED REINFORCEMENT LEARNING

Hyungkyu Kang
Seoul National University
Seoul, South Korea
hyungkyu0119@snu.ac.kr

Min-hwan Oh
Seoul National University
Seoul, South Korea
minoh@snu.ac.kr

ABSTRACT

In this paper, we study offline preference-based reinforcement learning (PbRL), where learning is based on pre-collected preference feedback over pairs of trajectories. While offline PbRL has demonstrated remarkable empirical success, existing theoretical approaches face challenges in ensuring conservatism under uncertainty, requiring computationally intractable confidence set constructions. We address this limitation by proposing Adversarial Preference-based Policy Optimization (APPO), a computationally efficient algorithm for offline PbRL that guarantees sample complexity bounds without relying on explicit confidence sets. By framing PbRL as a two-player game between a policy and a model, our approach enforces conservatism in a tractable manner. Using standard assumptions on function approximation and bounded trajectory concentrability, we derive a sample complexity bound. To our knowledge, APPO is the first offline PbRL algorithm to offer both statistical efficiency and practical applicability. Experimental results on continuous control tasks demonstrate that APPO effectively learns from complex datasets, showing comparable performance with existing state-of-the-art methods.

1 INTRODUCTION

While Reinforcement Learning (RL) has achieved remarkable success in real-world applications (Mnih, 2013; Silver et al., 2017; Kalashnikov et al., 2018; Brohan et al., 2022), its performance heavily depends on the design of the reward function (Wirth et al., 2017), which can be challenging in practice. To address this issue, preference-based reinforcement learning (PbRL), also known as reinforcement learning with human feedback, has gained increasing attention as an alternative to manually designed rewards. In PbRL, a reward model is learned from preference feedback provided by human experts, who compare pairs of trajectories (Christiano et al., 2017). This approach enables the learning process to align better with human intentions. PbRL has demonstrated its effectiveness in various domains, including gaming (MacGlashan et al., 2017; Christiano et al., 2017; Warnell et al., 2018), natural language processing (Ziegler et al., 2019; Stiennon et al., 2020; Nakano et al., 2021; Ouyang et al., 2022; Bai et al., 2022), and robotics (Brown et al., 2019; Shin et al., 2023).

However, collecting preference feedback can be costly, especially when real-time feedback from human experts is required. In such cases, learning from pre-collected data is preferred over online learning. This approach is referred to as *offline* PbRL, where the learning process relies solely on pre-collected trajectories and preference feedback. Empirical studies have shown the effectiveness of offline PbRL (Kim et al., 2023; An et al., 2023; Shin et al., 2023; Hejna & Sadigh, 2024), leveraging techniques from deep RL literature. On the theoretical side, prior works prove that trajectory concentrability with respect to the data-collecting distribution leads to a sample complexity bound (Zhu et al., 2023; Zhan et al., 2024a; Pace et al., 2024). However, they rely on the explicit construction of confidence sets to achieve conservatism (pessimism). Dealing with such confidence sets in the general function approximation setting requires intractable optimizations: Zhan et al. (2024a) involve tri-level constrained optimization with respect to the confidence sets of rewards and transitions, Pace et al. (2024) use uncertainty penalty defined as the width of confidence sets, and the analysis of Zhu et al. (2023) is restricted to linear models. Despite provable sample complexity bounds, existing offline PbRL algorithms become computationally intractable with general function approximation.

In this work, we propose a computationally and statistically efficient offline PbRL algorithm, *Adversarial Preference-based Policy Optimization* (APPO). Our analysis is based on general function approximation for both the model and value function classes. Moreover, standard assumptions on function classes and bounded trajectory concentrability (Zhan et al., 2024a) are sufficient to establish our sample complexity bound. Beyond its strong statistical guarantees, our algorithm is simple to implement using standard optimization techniques. The core idea behind our algorithm is the two-player game formulation of model-based PbRL, which has been used in other areas of RL (Rajeswaran et al., 2020; Rigter et al., 2022; Cheng et al., 2022; Shen et al., 2024; Bhardwaj et al., 2024). By framing PbRL as a game between a policy and a model, we ensure conservatism without explicitly constructing intractable confidence sets. Furthermore, our novel reparameterization technique allows us to find a near-optimal policy efficiently via adversarial training. To the best of our knowledge, APPO is the first offline PbRL algorithm with both statistical performance guarantees and a practical implementation. Our contributions can be summarized as follows:

- We propose APPO, a simple algorithm for offline PbRL with general function approximation. Based on the two-player game formulation of PbRL in conjunction with our reparameterization technique for the reward model, our algorithm ensures provable conservatism without explicit construction of confidence sets. To our knowledge, our APPO is the first computationally efficient offline PbRL algorithm providing a sample complexity bound.
- We prove the sample complexity of our proposed algorithm under standard assumptions on the function classes and concentrability. The result is rooted in our novel sub-optimality decomposition, which shows that adversarial training leads to model conservatism.
- We present a practical implementation of APPO that can learn with large datasets using neural networks. Experiments on continuous control tasks demonstrate that APPO achieves performance comparable to existing state-of-the-art algorithms.

1.1 Related Work

Provable Online PbRL. In the tabular setting, Novoseller et al. (2020) developed an algorithm grounded in posterior sampling and the dueling bandit framework (Yue et al., 2012), demonstrating an asymptotic rate for Bayesian regret. Xu et al. (2020) proposed an algorithm leveraging an exploration bonus for previously unseen states, which provides a sample complexity bound. Saha et al. (2023) and Zhan et al. (2024b) focused on the linear preference model with a known linear feature map, each offering regret and sample complexity bounds. However, their algorithms require solving an optimization $\arg\max_{\pi,\pi'} \|\mathbb{E}_{\tau\sim\pi}[\phi(\tau)] - \mathbb{E}_{\tau\sim\pi'}[\phi(\tau)]\|_{\Sigma}$ for some positive definite matrix Σ, which is computationally intractable. To address this challenge in the linear model, Wu & Sun (2024) devised a randomized algorithm with a provable regret bound and further proposed a model-based posterior sampling algorithm under the bounded Eluder dimension (Russo & Van Roy, 2013) assumption, ensuring bounded Bayesian regret. Recent works have also explored provably efficient algorithms under the general function approximation setting (Chen et al., 2022; Wu & Sun, 2024; Chen et al., 2023). Chen et al. (2022) introduced an exploration-bonus-based algorithm that provides bounded regret in both pairwise and n-wise comparison settings. Additionally, Chen et al. (2023) leveraged the conditional value-at-risk operator (Artzner, 1997) to devise an algorithm with a regret guarantee. Du et al. (2024) took a different approach, studying neural function approximation in the context of reward models. In another notable work, Swamy et al. (2024) reframed PbRL as a zero-sum game between two policies, encompassing general reward models.

Provable Offline PbRL. While there has been a growing amount of research on online PbRL, the theoretical understanding of offline PbRL remains relatively limited. A primary challenge in offline PbRL, much like in offline standard RL, is ensuring sufficient conservatism in the model. Zhu et al. (2023) addressed this challenge by proposing a pessimistic maximum likelihood estimation (MLE) algorithm for the linear model with known transitions. Zhan et al. (2024a) extended this idea to general function approximation, highlighting the importance of trajectory concentrability in establishing a lower bound for sample complexity. Despite the provable sample complexity bound, their proposed algorithm, FREEHAND-transition, relies on solving $\arg\max_{\pi} \arg\min_{r\in\hat{\mathcal{R}}} \arg\min_{P\in\hat{\mathcal{P}}} \{\mathbb{E}_{\tau\sim P,\pi}[r(\tau)] - \mathbb{E}_{\tau\sim P^\star,\pi}[r(\tau)]\}$ where $\hat{\mathcal{R}}$ is the confidence set of rewards and $\hat{\mathcal{P}}$ is the confidence set of transitions, which is intractable in practice. Pace et al. (2024) introduced an algorithm achieving conservatism through explicit un-

certainty penalties defined as $u_R(\tau) = \sup_{r_1, r_2 \in \hat{\mathcal{R}}} |r_1(\tau) - r_2(\tau)|$ (reward uncertainty) and $u_P(s, a) = \sup_{P_1, P_2 \in \hat{\mathcal{P}}} \|P_1(\cdot \mid s, a) - P_2(\cdot \mid s, a)\|_1$ (transition uncertainty). Even evaluating this function is intractable with general function approximation, but Pace et al. (2024) requires optimizing $\arg\max_\pi \mathbb{E}_{\tau \sim \hat{P}, \pi}[\hat{r}(\tau) - u_R(\tau) - u_P(\tau)]$. Chang et al. (2024) explored a slightly different scenario where the data collection policy is known and online interaction is allowed. They demonstrated that a simple natural policy gradient combined with MLE reward is provably efficient, but their sample complexity bound is affected by an additional concentrability coefficient relative to KL-regularized policies.

Adversarial Training in RL. Adversarial training is a widely used approach in RL literature (Rajeswaran et al., 2020; Pásztor et al., 2024), especially offline (standard) RL (Rigter et al., 2022; Cheng et al., 2022; Bhardwaj et al., 2024). The basic idea is to leverage adversarial training to implement conservative policy optimization. Recently, adversarial training has also been applied in human preference alignment (Makar-Limanov et al., 2024; Cheng et al., 2024; Shen et al., 2024). The most closely related work to ours is Shen et al. (2024), which also formulated PbRL as a two-player game. However, their focus is on online PbRL, and while they provide proof of convergence for the optimization objective, this does not necessarily translate into a sample complexity guarantee.

2 PRELIMINARIES

Markov Decision Processes. We consider an episodic MDP $(\mathcal{S}, \mathcal{A}, H, \{P_h^\star\}_{h=1}^H, \{r_h^\star\}_{h=1}^H)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state space and the action space, H is the length of each episode, $P^\star = \{P_h^\star\}_{h=1}^H$ is the collection of transition probability distributions, and $r^\star = \{r_h^\star\}_{h=1}^H$ is the collection of reward functions. Each episode starts at some initial state s_1 without loss of generality[1], and the episode ends after H steps. For each step $h \in [H]$, the agent observes the state s_h, and then takes action a_h. The environment generates reward $r_h^\star(s_h, a_h)$ (note that, in the preference-based learning setting, rewards at each step are unobservable to the agent) and next state s_{h+1} according to the transition probability $P_h^\star(\cdot \mid s_h, a_h)$.

The agent takes actions based on its policy $\pi = \{\pi_h\}_{h \in [H]}$, where $\pi_h(\cdot \mid s)$ is a probability distribution over $\mathcal{A}$. The state-value function and the action-value function of policy π with respect to reward $r = \{r_h\}_{h=1}^H$ are the expected sum of rewards up to termination, starting from $s_h = s$ and $(s_h, a_h) = (s, a)$ respectively, following the policy π. Formally, they are defined as

$$V_{h,r}^\pi(s) := \mathbb{E}_\pi \left[\sum_{h'=h}^H r_h(s_{h'}, a_{h'}) \mid s_h = s \right], \; Q_{h,r}^\pi := \mathbb{E}_\pi \left[\sum_{h'=h}^H r_h(s_{h'}, a_{h'}) \mid s_h = s, a_h = a \right].$$

To simplify the notation, for $g : \mathcal{S} \mapsto \mathbb{R}$, we use $Pg(s, a)$ to denote $\mathbb{E}_{s' \sim P(\cdot \mid s, a)}[g(s')]$. For any policy π and reward r, the Bellman equation relates Q^π to V^π as

$$Q_{h,r}^\pi(s, a) = r_h(s, a) + P^\star V_{h+1,r}^\pi(s, a), \; V_{h,r}^\pi(s) = \mathbb{E}_{a \sim \pi_h(\cdot \mid s)}[Q_{h,r}^\pi(s, a)], \; V_{H+1}^\pi(s) = 0.$$

Given a policy $\pi = \{\pi_h\}_{h \in [H]}$, we define the state visitation distribution as $d_h^\pi(s) := \mathbb{P}_\pi(s_h = s)$ where $\mathbb{P}_\pi$ is the probability distribution of trajectories $(s_1, a_1, \ldots, s_H, a_H)$ when the agent uses policy π. We overload the notation to denote the state-action visitation distribution, $d_h^\pi(s, a) := \mathbb{P}_\pi(s_h = s, a_h = a)$. In addition, we denote the distribution of trajectories under π by $d^\pi(\tau)$.

Offline Preference-based Reinforcement Learning. We consider the offline PbRL problem, where the agent cannot observe the true reward $r^\star$ but only binary preference feedback over trajectory pairs. Specifically, we are given a preference dataset $\mathcal{D}_{\text{pref}} = \{(\tau^{m,0}, \tau^{m,1}, y^m)\}_{m=1}^M$ that consists of i.i.d. trajectory pairs $\tau^{m,i} = \{s_h^{m,i}, a_h^{m,i}\}_{h=1}^H$ $(i = 0, 1)$ sampled by some reference policy μ. For a monotonically increasing link function $\Phi : \mathbb{R} \mapsto [0, 1]$, we assume the preference feedback $y^m \in \{0, 1\}$ is generated by the following preference model:

$$\mathbb{P}(y = 1 \mid \tau^0, \tau^1) = \mathbb{P}(\tau^1 \text{ is preferred over } \tau^0) = \Phi(r^\star(\tau^1) - r^\star(\tau^0))$$

where we denote $r^\star(\tau) = \sum_{h=1}^H r_h^\star(s_h, a_h)$ for given trajectory $\tau = (s_1, a_1, \ldots, s_H, a_H)$. Additionally, we assume that $\kappa = 1/(\inf_{x \in [-R, R]} \Phi'(x))$, where R is a bound on trajectory returns, is

[1]Our result easily extends to the general case with an initial distribution $\rho(\cdot)$. We can modify the MDP by setting a fixed initial state s_1 and $\mathbb{P}_1(\cdot \mid s_1, a) = \rho(\cdot)$ for all $a \in \mathcal{A}$.

finite. When Φ is set to be the sigmoid function $\sigma(x) = 1/(1+\exp(-x))$, we obtain the widely used Bradely-Terry-Luce model (Bradley & Terry, 1952). In addition to the preference dataset, we have an unlabeled trajectory dataset $\mathcal{D}_{\text{traj}} = \{(\tau^{0,n}, \tau^{1,n})\}_{n=1}^N$ where the trajectory pairs are sampled i.i.d. by executing the reference policy μ. The agent's goal is to find an ϵ-optimal policy $\hat{\pi}$ with respect to the optimal policy $\pi^\star$, which satisfies $V_{1,r^\star}^{\pi^\star}(s_1) - V_{1,r^\star}^{\hat{\pi}}(s_1) \leq \epsilon$.

General Function Approximation. We consider general function approximation for rewards and transitions: the function class of rewards $\mathcal{R}$ and the function class of transitions $\mathcal{P}$. We do not impose any specific structure on them, so $\mathcal{R}$ and $\mathcal{P}$ can contain expressive functions such as neural networks. Based on the function classes, we construct a reward model using maximum likelihood estimation $\hat{r} \in \arg\min_{r\in\mathcal{R}^H} \hat{\mathcal{L}}_R(r)$ where

$$\hat{\mathcal{L}}_R(r) = -\mathop{\mathbb{E}}_{(\tau^0,\tau^1,y)\sim\mathcal{D}_{\text{pref}}} \left[\mathbb{1}\{y=1\}\cdot\log\Phi(r(\tau^1)-r(\tau^0)) + \mathbb{1}\{y=0\}\cdot\log\Phi(r(\tau^0)-r(\tau^1))\right].$$

Similarly, we learn a transition model $\hat{P}_h \in \arg\min_{P\in\mathcal{P}} \hat{\mathcal{L}}_T(P;h)$ for all $h\in[H]$, where

$$\hat{\mathcal{L}}_T(P;h) = \mathbb{E}_{(s_h,a_h,s_{h+1})\sim\mathcal{D}_{\text{traj}}}\left[\log P(s_{h+1} \mid s_h, a_h)\right]$$

Additional Notations. We denote $[n] := \{1,2,\ldots,n\}$ for $n\in\mathbb{N}$. For $x,y\in\mathbb{R}^d$, $\langle x,y\rangle$ denotes the inner product of x and y. Given a function $f:\mathcal{S}\times\mathcal{A}\mapsto\mathbb{R}$ and a policy π, we write $f\circ\pi(s) := \mathbb{E}_{a\sim\pi(\cdot|s)}[f(s,a)]$. For given dataset $\mathcal{D}$, we use $\mathbb{E}_{x\sim\mathcal{D}}[f(x)]$ to denote $\frac{1}{|\mathcal{D}|}\sum_{x\in\mathcal{D}} f(x)$.

3 Algorithm

3.1 PbRL as a Two-player Game

A previous study on model-based PbRL by Zhan et al. (2024a) proves that the following optimization problem yields a near-optimal policy $\hat{\pi}$, for an appropriately chosen constant ζ:

$$\hat{\pi} \in \arg\max_{\pi}\min_{r\in\hat{\mathcal{R}}}\left(V_{1,r}^{\pi}(s_1) - V_{1,r}^{\mu}(s_1)\right) \text{ where } \hat{\mathcal{R}} = \left\{r\in\mathcal{R}^H : \hat{\mathcal{L}}_R(r) \leq \hat{\mathcal{L}}_R(\hat{r}) + \zeta\right\}. \tag{1}$$

The minimization with respect to reward model $r\in\hat{\mathcal{R}}$ ensures conservatism, which is essential for a provable guarantee. However, the constrained optimization is intractable with general function approximation. To address this challenge, we formulate the model-based PbRL problem as a two-player Stackelberg game (Von Stackelberg, 2010) between the policy and the reward:

$$\begin{aligned}\hat{\pi} &\in \arg\max_{\pi}\left(V_{1,r^\pi}^{\pi}(s_1) - V_{1,r^\pi}^{\mu}(s_1)\right)\\ \text{subject to } r^\pi &\in \arg\min_{r\in\mathcal{R}^H}\left(V_{1,r}^{\pi}(s_1) - V_{1,r}^{\mu}(s_1) + \mathcal{E}(r;\hat{r})\right).\end{aligned} \tag{2}$$

Here, $\mathcal{E}(r;\hat{r})$ is a loss function that penalizes r if it deviates from $\hat{r}$. In the Stackelberg game formulation, the reward minimizes $V_{1,r}^{\pi}(s_1) - V_{1,r}^{\mu}(s_1)$, while the policy maximizes it. We can interpret this competition by viewing $V_{1,r}^{\pi}(s_1) - V_{1,r}^{\mu}(s_1)$ as the relative performance of π compared to μ with respect to reward r. Intuitively, π maximizes the cumulative reward r^π, as in the standard RL setup. However, r^π minimizes the cumulative reward when evaluated under π. This competition facilitates conservatism and makes π robust to model error.

Then, what loss function $\mathcal{E}$ leads to a provable bound? A naive choice might be $\hat{\mathcal{L}}_R(r) - \hat{\mathcal{L}}_R(\hat{r})$, as it leads to the Lagrangian dual form of the optimization problem in (1), while disregarding the Lagrangian multiplier. However, the loss $\mathcal{E}(r;\hat{r}) = \hat{\mathcal{L}}_R(r) - \hat{\mathcal{L}}_R(\hat{r})$ does not guarantee statistical efficiency, because the Stackelberg game in (2) does not include the Lagrangian multiplier for the likelihood constraint. Instead, we propose the trajectory-pair ℓ_1 loss:

$$\mathcal{E}(r;\hat{r}) = \mathbb{E}_{\tau^0,\tau^1\sim\mu}\left[\left|\{r(\tau^0)-r(\tau^1)\} - \{\hat{r}(\tau^0)-\hat{r}(\tau^1)\}\right|\right],$$

which leads to a provable guarantee (Theorem 4.1). Intuitively, this loss measures the deviation of r from $\hat{r}$ by evaluating the difference in total reward (return) between the two trajectories. Given the unlabeled trajectory dataset $\mathcal{D}_{\text{traj}}$, we approximate $\mathcal{E}(r;\hat{r})$ with its finite-sample version:

$$\hat{\mathcal{E}}_{\mathcal{D}_{\text{traj}}}(r;\hat{r}) = \mathbb{E}_{(\tau^0,\tau^1)\sim\mathcal{D}_{\text{traj}}}\left[\left|\{r(\tau^0)-r(\tau^1)\} - \{\hat{r}(\tau^0)-\hat{r}(\tau^1)\}\right|\right].$$

In the following two sections, we discuss how to implement the optimization in (2) in a sample-efficient manner.

Algorithm 1 Adversarial Preference-based Policy Optimization with Rollout (`APPO-rollout`)

1: **Input:** Number of rollouts K_1, K_2, KL regularization η, $\pi_h^1 = \text{Unif}(\mathcal{A})$ for all $h \in [H]$
2: Estimate $\hat{r} \in \arg\min_{r \in \mathcal{R}^H} \hat{\mathcal{L}}_R(r)$
3: **for** $t = 1, \cdots, T$ **do**
4: Execute π^t to collect K_1 trajectories $\mathcal{D}^t_{\text{rollout}}$
5: Optimize $r^t \in \arg\min_{r \in \mathcal{R}^H} \left(\mathbb{E}_{\tau \sim \mathcal{D}^t_{\text{rollout}}}[r(\tau)] - \mathbb{E}_{\tau \sim \mathcal{D}_{\text{traj}}}[r(\tau)] + \lambda \hat{\mathcal{E}}_{\mathcal{D}_{\text{traj}}}(r; \hat{r}) \right)$
6: Compute $\bar{Q}^t$ via $\texttt{PE}(\mu, \pi^t, \hat{r}, K_2)$ in Algorithm 3
7: Update policy $\pi_h^{t+1}(a \mid s) \propto \pi_h^t(a \mid s) \exp(\eta \bar{Q}_h^t(s,a))$ for all $h \in [H]$
8: **end for**
9: Return $\bar{\pi} = \frac{1}{T} \sum_{t=1}^T \pi_t$

Algorithm 2 Adversarial Preference-based Policy Optimization (`APPO`)

1: **Input:** KL regularization η, Initial policy $\pi_h^1 = \text{Unif}(\mathcal{A})$ for all $h \in [H]$
2: Estimate $\hat{r} \in \arg\min_{r \in \mathcal{R}^H} \hat{\mathcal{L}}_R(r)$, $\hat{P}_h \in \arg\min_{P \in \mathcal{P}} \hat{\mathcal{L}}_T(P; h)$ for all $h \in [H]$
3: **for** $t = 1, \cdots, T$ **do**
4: $f^t \in \arg\min_{f \in \mathcal{F}^H} \left(\sum_{h=1}^H \mathbb{E}_{(s_h, a_h) \sim \mathcal{D}_{\text{traj}}} [f_h \circ \pi_h^t(s_h) - f_h(s_h, a_h)] + \lambda \hat{\mathcal{E}}_{\mathcal{D}_{\text{traj}}}(f; \hat{P}, \hat{r}) \right)$
5: Update policy $\pi_h^{t+1}(a \mid s) \propto \pi_h^t(a \mid s) \exp(\eta f_h^t(s,a))$ for $h \in [H]$
6: **end for**
7: Return $\hat{\pi} = \frac{1}{T} \sum_{t=1}^T \pi^t$

3.2 Adversarial Optimization for PbRL

In this section, we present an algorithm, `APPO-rollout`, that serves as a building block of our main algorithm. For `APPO-rollout`, we consider the setting where the transition $P^\star$ is known or where online interaction (without preference feedback) is possible. This is a temporary assumption, as our main algorithm (Algorithm 2) works with an unknown transition.

Algorithm 1 presents the pseudo-code of `APPO-rollout`, which is based on the Stackelberg game formulation of PbRL that we discussed. Inspired by the adversarial training methods in offline RL under the standard setting (Cheng et al., 2022; Rigter et al., 2022; Bhardwaj et al., 2024), we alternately optimize the policy and reward to solve the optimization problem in (2).

Reward Model Update for Provable Conservatism. The reward model update aims to solve the following optimization problem approximately:

$$\arg\min_{r \in \mathcal{R}^H} \left(\mathop{\mathbb{E}}_{\tau \sim \pi^t}[r(\tau)] - \mathop{\mathbb{E}}_{\tau \sim \mu}[r(\tau)] + \lambda \mathcal{E}(r; \hat{r}) \right) = \arg\min_{r \in \mathcal{R}^H} \left(V_{1,r}^{\pi^t}(s_1) - V_{1,r}^{\mu}(s_1) + \lambda \mathcal{E}(r; \hat{r}) \right), \quad (3)$$

which is the inner optimization in (2). The expectations $\mathbb{E}_{\tau \sim \mu}[r(\tau)]$ and $\mathcal{E}(r; \hat{r})$ are approximated using offline data $\mathcal{D}_{\text{traj}}$. Also, we collect trajectories by executing π^t, to compute the finite-sample version of $\mathbb{E}_{\tau \sim \pi^t}[r(\tau)]$. Note that the trajectory rollout (Line 4) is possible since we assume a known transition $P^\star$ or access to online interaction.

Policy Update. After optimizing r^t, we estimate the action-value function of π^t with respect to r^t using a policy evaluation subroutine `PE`, whose pseudo-code is provided in Algorithm 3. This subroutine computes an approximate value function $\bar{Q}^t$ using Monte Carlo estimation, providing an error bound relative to the true value function $Q_{r^t}^{\pi^t}$. The theoretical analysis of `PE` is presented in Appendix B. With the estimated value function $\bar{Q}^t$, we then proceed to update the policy using trust region policy optimization (TRPO) (Schulman et al., 2015) update.

3.3 `APPO`: Reparameterized Algorithm for Unknown Transition

In this section, we consider the setting where the transition $P^\star$ is unknown. In Algorithm 1, the information from the transition $P^\star$ is utilized in Line 4, where we collect on-policy trajectories to approximate $\mathbb{E}_{\tau \sim \pi^t}[r(\tau)]$. Moreover, the policy evaluation step in Algorithm 3 involves trajectory

rollouts. To bypass such on-policy rollouts, we make the following observation:

$$\begin{aligned}\mathbb{E}_{\tau\sim\pi^t}[r(\tau)] - \mathbb{E}_{\tau\sim\mu}[r(\tau)] &= V_{1,r}^{\pi^t}(s_1) - V_{1,r}^{\mu}(s_1) \\ &= \sum_{h=1}^{H} \mathbb{E}_{(s_h,a_h)\sim d_h^\mu}\left[(Q_{h,r}^{\pi^t}\circ\pi_h^t)(s_h) - Q_{h,r}^{\pi^t}(s_h,a_h)\right] \end{aligned} \tag{4}$$

which is due to the performance difference lemma (Lemma E.1). Since the expectation on the right is taken with respect to d_h^μ, the data-generating distribution of $\mathcal{D}_{\text{traj}}$, we can approximate it using $\mathcal{D}_{\text{traj}}$. Furthermore, given the policy π^t, the Bellman equation implies a mapping between reward models and action-value functions. Specifically, for given reward model $r = \{r_h\}_{h=1}^H$, we have the action-value function $\{Q_{h,r}^{\pi^t}\}_{h=1}^H$. Conversely, suppose that we have a function class $\mathcal{F}$ that contains every action-value function. For $f = \{f_h\}_{h=1}^H \in \mathcal{F}^H$, we can construct the corresponding reward model satisfying the Bellman equation $f_h = r_h + P_h^\star(f_{h+1}\circ\pi_{h+1}^t)$. Formally, we define the induced reward models:

Definition 1 (Induced reward model)**.** *Given $f = \{f_h\}_{h=1}^H \in \mathcal{F}^H$, and a policy $\{\pi_h\}_{h=1}^H$, we define the induced reward model $r_{P^\star,f}^\pi = \{r_{h,P^\star,f}^\pi\}_{h=1}^H$ where $r_{h,P^\star,f}^\pi = f_h - P_h^\star(f_{h+1}\circ\pi_{h+1})$ for $h\in[H]$ (we set $f_{H+1}=0$ by convention).*

Therefore, given reward model r and action-value function f, we have that

$$Q_{h,r}^\pi = r_h + P^\star(Q_{h+1,r}^\pi\circ\pi_{h+1}), \quad f_h = r_{h,P^\star,r}^\pi + P_h^\star(f_{h+1}\circ\pi_h) \text{ for all } h\in[H].$$

Importantly, *the mapping does not need to be bijective* for our theoretical analysis, as long as the Bellman equation holds. Using this mapping in conjunction with our observation in (4), we reparameterize the optimization problem in (3) as:

$$\begin{aligned}&\arg\min_{f\in\mathcal{F}^H}\left(\sum_{h=1}^{H}\mathbb{E}_{(s_h,a_h)\sim d_h^\mu}\left[(f_h\circ\pi_h^t)(s_h) - f_h(s_h,a_h)\right] + \lambda\mathcal{E}(f;P^\star,\hat{r})\right) \\ &\text{where } \mathcal{E}(f;P^\star,\hat{r}) = \mathbb{E}_{(\tau_0,\tau_1)\sim\mu}\left[\left|\{r_{P^\star,f}^{\pi^t}(\tau^0) - r_{P^\star,f}^{\pi^t}(\tau^1)\} - \{\hat{r}(\tau^0)-\hat{r}(\tau^1)\}\right|\right].\end{aligned} \tag{5}$$

The offline dataset $\mathcal{D}_{\text{traj}}$ is sufficient to approximate the optimization objective in (5) with

$$\begin{aligned}&\mathbb{E}_{(s_h,a_h)\sim\mathcal{D}_{\text{traj}}}\left[(f_h\circ\pi_h^t)(s_h) - f_h(s_h,a_h)\right] \approx \mathbb{E}_{(s_h,a_h)\sim d_h^\mu}\left[(f_h\circ\pi_h^t)(s_h) - f_h(s_h,a_h)\right] \\ &\hat{\mathcal{E}}_{\mathcal{D}_{\text{traj}}}(f;\hat{P},\hat{r}) := \mathbb{E}_{(\tau_0,\tau_1)\sim\mu}\left[\left|\{r_{\hat{P},f}^{\pi^t}(\tau^0) - r_{\hat{P},f}^{\pi^t}(\tau^1)\} - \{\hat{r}(\tau^0)-\hat{r}(\tau^1)\}\right|\right] \approx \mathcal{E}(f;P^\star,\hat{r}),\end{aligned}$$

where we use the estimated transition model $\hat{P}$ in place of $P^\star$. Moreover, since we directly optimize the action-value function, a policy evaluation oracle is not required to update the policy. Therefore, this reparameterization allows us to solve the optimization problem in (2) without access to the true transition $P^\star$ or a policy evaluation oracle. The complete pseudo-code is presented in Algorithm 2.

Remark on Computational Complexity. The computational complexity of `APPO` is primarily determined by the value function optimization (Line 4) and the policy update (Line 5). Although optimizing f^t is generally a non-convex problem, it can be efficiently implemented using gradient-based methods when $\mathcal{F}$ is a class of neural networks. For the policy update, it is known that $\pi_h^{t+1}(a \mid s) \propto \pi_h^t(a \mid s)\exp(\eta f_h^t(s,a))$ is derived from the TRPO objective (Schulman et al., 2015; Neu et al., 2017):

$$\pi_h^{t+1} \in \arg\max_\pi \mathbb{E}_{s_h\sim d_h^{\pi^t}}\left[f_h^t\circ\pi(s_h) - \eta^{-1}D_{KL}\left(\pi(\cdot\mid s_h)\|\pi_h^t(\cdot\mid s_h)\right)\right],$$

which is widely used in deep RL. As a result, the policy update is efficient within the deep learning framework. In practice, other policy optimization techniques (Schulman et al., 2017; Fujimoto et al., 2018; Haarnoja et al., 2018) can also be applied. Overall, `APPO` relies on solving two standard non-convex optimizations to compute f^t and π^t, both of which are practical to implement with neural function approximation. This computational efficiency contrasts with that of existing offline PbRL algorithms, which require intractable optimization over confidence sets, as discussed in Section 1.1.

4 THEORETICAL ANALYSIS

In this section, we present theoretical analyses of our proposed algorithm, `APPO`. We note that `APPO-rollout` also guarantees a sample complexity bound, which is presented in Appendix C.

We assume the reward class $\mathcal{R}$ and the transition class $\mathcal{P}$ are realizable and rewards are bounded. These are standard assumptions (Chen et al., 2023; Zhan et al., 2024a; Pace et al., 2024).

Assumption 1 (Reward realizability)**.** *We have $r_h^\star \in \mathcal{R}$ for all $h \in [H]$. In addition, every $r \in \mathcal{R}^H$ satisfies $0 \leq r(\tau) \leq R$ for any trajectory τ.*

Assumption 2 (Transition realizability)**.** *We have $P_h^\star \in \mathcal{P}$ for all $h \in [H]$.*

Additionally, we introduce the value function class and assume it is bounded. Note that every $Q_{h,r}^\pi$ satisfies the condition $\|f\|_\infty \leq R$ due to Assumption 1.

Assumption 3 (Value function class)**.** *For any $h \in [H]$, $r \in \mathcal{R}^H$, and policy π, we have $Q_{h,r}^\pi \in \mathcal{F}$. In addition, every $f \in \mathcal{F}$ satisfies $0 \leq f(s,a) \leq R$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$.*

The following assumption defines the trajectory concentrability coefficient between the optimal policy $\pi^\star$ and the reference policy μ.

Assumption 4 (Trajectory concentrability)**.** *There exists a finite constant C_{TR} such that the behavior policy μ and the optimal policy $\pi^\star$ satisfy $\sup_\tau \frac{d^{\pi^\star}(\tau)}{d^\mu(\tau)} \leq C_{TR}$.*

The bounded C_{TR} ensures that the support of d^μ sufficiently covers the support of $d^{\pi^\star}$, similar to the concentrability condition in Zhan et al. (2024a). As a result, we expect $\mathcal{D}_{\text{traj}}$ to contain high-quality trajectories. The lower bound in Zhan et al. (2024a) shows that the trajectory concentrability is essential in offline PbRL. This implies that offline PbRL is strictly more challenging than offline standard RL, where step-wise concentrability is sufficient to achieve a performance guarantee (Uehara & Sun, 2022). We now present the sample complexity bound.

Theorem 4.1. *Suppose Assumptions 1,2, 3, and 4 hold. With probability at least $1-\delta$, Algorithm 2 with $\lambda = \Theta(C_{TR}), \lambda > C_{TR}, \eta = \sqrt{\frac{2\log|\mathcal{A}|}{R^2 T}}$ achieves*

$$V_{1,r^\star}^{\pi^\star} - V_{1,r^\star}^{\hat{\pi}} \leq \mathcal{O}\left(C_{TR}\sqrt{\frac{\kappa^2 H}{M}\log\frac{|\mathcal{R}|}{\delta}} + RH\sqrt{\frac{1}{N}\max\left\{HT\log\frac{H|\mathcal{F}|}{\delta}, \log\frac{H|\mathcal{P}|}{\delta}\right\}} + RH\sqrt{\frac{\log|\mathcal{A}|}{T}}\right).$$

Setting $T = \Theta\left(\frac{R^2H^2\log|\mathcal{A}|}{\epsilon^2}\right)$, $N = \Theta\left(\max\left\{\frac{R^4H^5\log|\mathcal{A}|\log(H|\mathcal{F}|/\delta)}{\epsilon^4}, \frac{R^2H^2\log(H|\mathcal{P}|/\delta)}{\epsilon^2}\right\}\right)$, and $M = \Theta\left(\frac{C_{TR}^2\kappa^2 H\log(|\mathcal{R}|/\delta)}{\epsilon^2}\right)$, Algorithm 2 achieves ϵ-optimal policy, i.e. $V_{1,r^\star}^{\pi^\star} - V_{1,r^\star}^{\bar{\pi}} \leq \epsilon$.

Discussion on Theorem 4.1. Our analysis naturally extends to infinite function classes by applying the standard covering number argument, replacing the cardinalities $|\mathcal{R}|$, $|\mathcal{P}|$, and $|\mathcal{F}|$ with covering numbers. To our knowledge, FREEHAND-transition Zhan et al. (2024a) and Sim-OPRL Pace et al. (2024) are the only statistically efficient algorithms for offline PbRL in stochastic MDPs. Our sample complexity bound matches theirs for labeled data (M). However, FREEHAND-transition and Sim-OPRL require $\Theta\left(\frac{C_P^2 R^2 H^2\log(H|\mathcal{P}|/\delta)}{\epsilon^2}\right)$ unlabeled trajectories where C_P is the trajectory concentrability for transition [2]. This highlights a trade-off: While FREEHAND-transition and Sim-OPRL have tighter bounds for unlabeled data (N), they are computationally intractable. FREEHAND-transition requires solving a nearly intractable nested optimization problem, while Sim-OPRL relies on an uncertainty penalty defined by the width of confidence sets. Therefore, our `APPO` is the first offline PbRL algorithm to achieve both provable statistical efficiency and computational efficiency.

[2] Zhan et al. (2024a) consider reward functions defined over trajectories, thus their reward class $\mathcal{G}_r$ is comparable to our $\mathcal{R}^H$. Pace et al. (2024) assume a homogeneous reward, so their presented bound is tighter by H. They use bracketing numbers in their bound, but we write here $|\mathcal{P}|$ for simplicity.

Proof Sketch. We outline the proof of Theorem 4.1, where the detailed proof is deferred to Appendix D. The key observation is our novel sub-optimality decomposition:

$$V^{\pi^\star}_{1,r^\star} - V^{\pi^t}_{1,r^\star} = \underbrace{V^{\pi^\star}_{1,r^\star-\hat{r}} - V^{\mu}_{1,r^\star-\hat{r}}}_{\text{(I) : MLE error}} + \underbrace{V^{\pi^\star}_{1,\hat{r}-r^t} - V^{\mu}_{1,\hat{r}-r^t} - V^{\pi^t}_{1,r^\star} + V^{\mu}_{1,r^\star} + V^{\pi^t}_{1,r^t} - V^{\mu}_{1,r^t}}_{\text{(II) : Optimization error}} + \underbrace{V^{\pi^\star}_{1,r^t} - V^{\pi^t}_{1,r^t}}_{\text{(III) : Policy update regret}},$$

where $r^t = r^{\pi^t}_{P^\star,f^t}$, and the initial state s_1 is omitted here for readability. The term (I) is bounded by a standard MLE guarantee (Lemma E.2), while the policy update rule ensures that the sum of terms (III) over T steps is bounded (Lemma D.3). For (II), Assumption 4 and $\lambda > C_{\text{TR}}$ imply that

$$\begin{aligned} V^{\pi^\star}_{1,\hat{r}-r^t} - V^{\mu}_{1,\hat{r}-r^t} &= \mathbb{E}_{\tau^0\sim\pi^\star,\tau^1\sim\mu}\left[r^t(\tau^0) - \hat{r}(\tau^0) - r^t(\tau^1) + \hat{r}(\tau^1)\right] \\ &\le C_{\text{TR}}\mathbb{E}_{\tau^0\sim\pi^\star,\tau^1\sim\mu}\left[\left|r^t(\tau^0) - \hat{r}(\tau^0) - r^t(\tau^1) + \hat{r}(\tau^1)\right|\right] \le \lambda\mathcal{E}(f^t;P^\star,\hat{r}). \end{aligned}$$

Observe that APPO approximately solves the optimization problem in (5) (Lemma D.1), and this optimization problem is equivalent to $\arg\min_{f\in\mathcal{F}^H}\{V^{\pi^t}_{1,r^{\pi^t}_{P^\star,f}} - V^{\mu}_{1,r^{\pi^t}_{P^\star,f}} + \lambda\mathcal{E}(f;P^\star,\hat{r})\}$. Since $r^{\pi^t}_{P^\star,f^t} = r^t$ and $r^{\pi^t}_{P^\star,Q^{\pi^t}} = r^\star$, it follows that

$$V^{\pi^t}_{1,r^t} - V^{\mu}_{1,r^t} + \lambda\mathcal{E}(f^t;P^\star,\hat{r}) \le V^{\pi^t}_{1,r^\star} - V^{\mu}_{1,r^\star} + \lambda\mathcal{E}(Q^{\pi^t};P^\star,\hat{r}) + \epsilon.$$

where ϵ represents an approximation error. Thus, we obtain (II)$\le \epsilon$. Combining the results with $V^{\pi^\star}_{1,r^\star} - V^{\hat{\pi}}_{1,r^\star} = \frac{1}{T}\sum_{t=1}^{T}\left(V^{\pi^\star}_{1,r^\star} - V^{\pi^t}_{1,r^\star}\right)$, we complete the proof.

5 Practical Implementation of APPO

While providing strong statistical guarantees, APPO allows practical implementation with neural networks, leveraging advanced training techniques from deep learning. In this section, we present a practical version of APPO tailored for deep PbRL. The pseudo-code is outlined in Algorithm 4. For practical implementation, we assume the standard discounted MDP setting in deep PbRL (Christiano et al., 2017), where trajectory segments of length L are given and preference labels are assigned to segment pairs.

Reward Learning. While our theoretical analysis is based on the maximum likelihood estimator, any reward learning strategy can be employed. This flexibility allows APPO to benefit from state-of-the-arts preference learning methods, such as data augmentation (Park et al., 2022) and active query techniques (Shin et al., 2023; Hwang et al., 2024; Choi et al., 2024).

Training Value Functions. Given a parameterized policy π_θ and an action-value function Q_ϕ, the optimization objective in (5) can be adapted to the discounted setting as follows:

$$\arg\min_{\phi} \mathbb{E}_{(s,a)\sim d^\mu}\left[(Q_\phi\circ\pi_\theta)(s) - Q_\phi(s,a)\right] + \lambda\mathbb{E}_{(\tau_0,\tau_1)\sim\mu}\left[\left|(r^\theta_\phi - \hat{r})(\tau^0) - (r^\theta_\phi - \hat{r})(\tau^1)\right|\right]$$

where $r^\theta_\phi(\tau) = \sum_{l=1}^{L}\left(Q_\phi(s_l,a_l) - \gamma(Q_\phi\circ\pi_\theta)(s_{l+1})\right)$ for the segment $\tau = (s_1,a_1,\ldots,s_L,a_L)$. We employ the approximation $P^\star(Q_\phi\circ\pi_\theta)(s_l,a_l) \approx (Q_\phi\circ\pi_\theta)(s_{l+1})$ to avoid the need for a transition model. Additionally, to stabilize training, we apply the clipped double Q-learning trick (Fujimoto et al., 2018; Haarnoja et al., 2018) and maintain a separate value-function V_ψ. Given mini-batches of trajectory pairs $\mathcal{B}_{\text{traj}}$ and transition tuples $\mathcal{B}_{\text{tup}}$, each action-value function Q_{ϕ^i} is trained by minimizing $\mathcal{L}^\lambda_{\phi^i} = \lambda\mathcal{L}^{\text{adv}}_{\phi^i} + \mathcal{E}_{\phi^i}$ (where λ is moved to the first term, without loss of generality), defined as follows:

$$\begin{aligned} \mathcal{L}^{\text{adv}}_{\phi^i}(\mathcal{B}_{\text{tup}}) &= \mathbb{E}_{(s,a)\sim\mathcal{B}_{\text{tup}}}\left[Q_{\phi^i}(s,\pi_\theta(s)) - Q_{\phi^i}(s,a)\right], \\ \text{and } \mathcal{E}_{\phi^i}(\mathcal{B}_{\text{traj}}) &= \mathbb{E}_{(\tau^0,\tau^1)\sim\mathcal{B}_{\text{traj}}}\left[\left|\{r^\psi_{\phi^i}(\tau^0) - r^\psi_{\phi^i}(\tau^1)\} - \{\hat{r}(\tau^0) - \hat{r}(\tau^1)\}\right|\right]. \end{aligned} \tag{6}$$

Here, we use the notation $r^\psi_{\phi^i}(\tau) = \sum_{l=1}^{L}\left(Q_{\phi^i}(s_l,a_l) - \gamma V_\psi(s_{h+l})\right)$, and $\pi_\theta(s)$ denotes an action sampled from $\pi_\theta(\cdot\mid s)$. Given target Q-networks $\{\bar{\phi}^i\}_{i\in\{1,2\}}$, we train V_ψ by minimizing

$$\mathcal{L}_\psi(\mathcal{B}_{\text{tup}}) = \mathbb{E}_{s\sim\mathcal{B}_{\text{tup}}}\left[\left(V_\psi(s) - \min_{i\in\{1,2\}} Q_{\bar{\phi}^i}(s_{h+1},\pi_\theta(s_{h+1}))\right)^2\right], \tag{7}$$

Dataset & # of feedback	Oracle	MR	PT	DPPO	IPL	APPO (ours)
BPT-500	88.33±4.76	10.08±7.57	22.87±9.06	3.93±4.34	34.73±13.9	**53.52**±13.9
box-close-500	93.40±3.10	**29.12**±13.2	0.33±1.16	10.20±11.5	5.93±5.81	**18.24**±15.6
dial-turn-500	75.40±5.47	61.44±6.08	**68.67**±12.4	26.67±22.2	31.53±12.5	**80.96**±4.49
sweep-500	98.33±1.87	**86.96**±6.93	43.07±24.6	10.47±15.8	27.20±23.8	26.80±5.32
BPT-wall-500	56.27±6.32	0.32±0.30	0.87±1.43	0.80±1.51	8.93±9.84	**64.32**±21.0
sweep-into-500	78.80±7.96	**28.40**±5.47	20.53±8.26	23.07±7.02	**32.20**±7.35	24.08±5.91
drawer-open-500	100.00±0.00	**98.00**±2.32	88.73±11.6	35.93±11.2	19.00±13.6	87.68±10.0
lever-pull-500	98.47±1.77	**79.28**±2.95	**82.40**±22.7	10.13±12.2	31.20±15.8	**75.76**±7.17
BPT-1000	88.33±4.76	8.48±5.80	18.27±10.6	3.20±3.04	36.67±17.4	**59.04**±19.0
box-close-1000	93.40±3.10	**27.04**±14.5	2.27±2.86	9.33±9.60	6.73±8.41	**34.24**±18.5
dial-turn-1000	75.40±5.47	69.44±4.70	68.80±5.50	36.40±21.9	43.93±13.4	**81.44**±6.73
sweep-1000	98.33±1.87	**87.52**±7.87	29.13±14.6	8.73±16.4	38.33±24.9	17.36±12.4
BPT-wall-1000	56.27±6.32	0.48±0.47	2.13±2.96	0.27±0.85	14.07±11.5	**62.96**±18.4
sweep-into-1000	78.80±7.96	**26.00**±5.53	20.27±7.84	**23.33**±7.80	**30.40**±7.74	18.16±11.1
drawer-open-1000	100.00±0.00	**98.40**±2.82	**95.40**±7.27	36.47±7.30	28.53±18.4	**98.56**±2.68
lever-pull-1000	98.47±1.77	**88.96**±3.94	72.93±10.2	8.53±9.96	40.40±17.4	76.96±4.40
Average Rank	-	2.316	3.125	4.375	3.063	**2.125**

Table 1: Success rates on Meta-World `medium-replay` dataset with 500 and 1000 preference feedback samples, averaged over 5 random seeds. The results of baselines, Oracle, PT, DPPO, and IPL, are taken from Choi et al. (2024), where Oracle refers to the policy trained using IQL with ground-truth rewards. The abbreviation BPT stands for button-press-topdown.

Intuitively, the term $\mathcal{L}_\phi^{\text{adv}}$ ensures conservatism by regularizing Q_ϕ to have lower values near d^{π_θ} and higher values near d^μ. Additional insight can be gained by rearranging the integrand of $\mathcal{E}_\phi$:

$$r_{\phi^i}^{\psi}(\tau) - \hat{r}(\tau) = \sum_{l=1}^{L} \left(Q_{\phi^i}(s_l, a_l) - \hat{r}(s_l, a_l) - \gamma V_\psi(s_{l+1}) \right) \right).$$

This expression represents the sum of the TD errors evaluated over the segment τ. Thus, the loss $\mathcal{E}_\phi$ minimizes the difference in trajectory TD errors between τ^0 and τ^1.

Training Policy. We directly optimize the policy using the loss function in (8). The entropy regularization term is similar to that in SAC (Haarnoja et al., 2018), except that we use a randomly sampled Q_{ϕ^i} instead of the clipped value $\min_{i\in[1,2]} Q_{\phi^i}$. The policy loss is given by:

$$\mathcal{L}_\theta(\mathcal{B}_{\text{tup}}) = \mathbb{E}_{s\sim\mathcal{B}_{\text{tup}}} \left[Q_{\phi^i}(s, \pi_\theta(s)) - \alpha \pi_\theta(s, \pi_\theta(s)) \right], \; i \sim \text{Unif}\{1, 2\} \tag{8}$$

6 Experiments

Datasets and Evaluation. We evaluate our proposed algorithm on the Meta-World (Yu et al., 2020) `medium-replay` and `medium-expert` datasets from Choi et al. (2024). Our main experiments use the `medium-replay` dataset, while the experiments with the `medium-expert` dataset are presented in Appendix F. These datasets have a favorable property: they are not learnable with incorrect rewards (random or constant). This property is crucial for evaluating offline RL algorithms since their survival instinct can allow them to perform well even with completely incorrect reward signals (Li et al., 2024). For more details on the dataset, see Choi et al. (2024). Following the experiment protocol of Choi et al. (2024), the preference dataset consists of pairs of randomly sampled trajectory segments of length 25. The preference label is generated based on the ground truth reward, where a $(0, 1)$ label is assigned if the trajectory rewards differ by more than a threshold of 12.5, and a $(0.5, 0.5)$ label is assigned otherwise. We evaluate algorithm performance using the success rate for each task, which indicates whether the agent successfully completes the task.

Algorithms. We consider four offline PbRL algorithms as baselines: Markovian Reward (MR), Preference Transformer (PT) (Kim et al., 2023), Direct Preference-based Policy Optimization

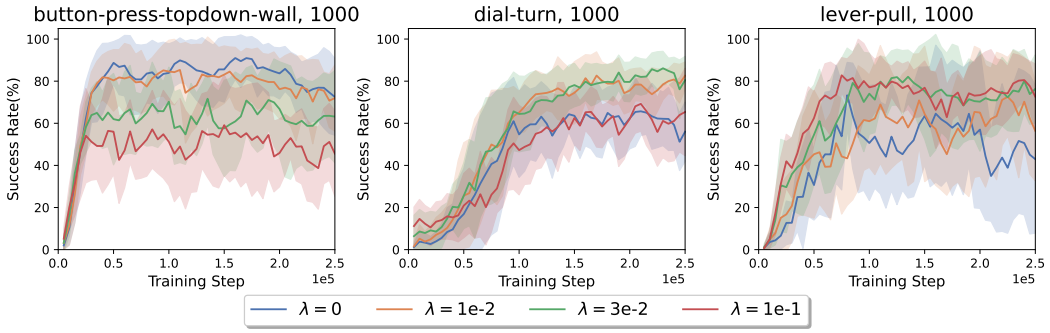

Figure 1: Effect of the conservatism regularizer λ.

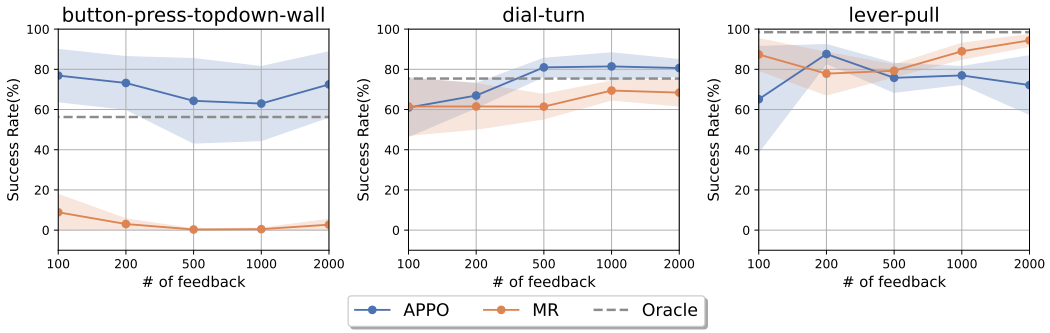

Figure 2: Success rates of APPO and MR, with varying number of preference feedback samples.

(DPPO) (An et al., 2023), and Inverse Preference Learning (IPL) (Hejna & Sadigh, 2024). MR is an instance of IQL (Kostrikov et al., 2022) trained with a Markovian reward model, while PT assumes a general sequential reward model implemented using a Transformer (Vaswani, 2017) architecture. DPPO directly optimizes the policy without using a reward model, while the other baseline methods are based on IQL (Kostrikov et al., 2022). We evaluate the practical version of APPO in Algorithm 4, using the same reward model as MR and setting $\lambda = 0.03$. Further details are in Appendix G.

6.1 Evaluation Results

Table 1 shows the performance of algorithms on the Meta-World control tasks. APPO outperforms or shows comparable performance in almost every dataset. Notably, APPO outperforms the policy trained with ground truth rewards in the dial-turn and button-press-topdown-wall datasets. We also observe that MR is a strong baseline, as reported in previous works (Hejna & Sadigh, 2024; Choi et al., 2024). These results suggest that APPO performs comparably to state-of-the-art baselines, even with a provable statistical guarantee.

Effect of Conservatism Regularizer. We investigate the effect of conservatism regularizer λ, the coefficient to balance the adversarial loss $\mathcal{L}_\phi^{\text{adv}}$ and the trajectory-pair ℓ_1 loss $\mathcal{E}_\phi$. In Figure 1, APPO learns successfully with a wide range of λ, but a properly tuned λ improves performance and stability. We note that APPO has only one algorithmic hyperparameter λ, whereas IQL-based algorithms (MR, PT, IPL) require at least two (expectile parameter and temperature). DPPO, on the other hand, has two hyperparameters (conservatism regularizer and smoothness regularizer).

Effect of Preference Dataset Size. In PbRL, learning from small preference datasets is desired for cost-efficient learning. We evaluate the effect of preference dataset size on APPO's performance, varying the number of feedback samples from 100 to 2000. Figure 2 shows that APPO is robust to preference dataset size, displaying variance comparable to MR, a strong baseline, as shown in Table 1. Note that APPO outperforms a policy trained with ground truth rewards, using only 100 preference feedback samples.

REPRODUCIBILITY

We describe the experimental details in Section 6 and Section G, including training protocol and neural network architecture. Our code is available at `https://github.com/oh-lab/APPO.git`. As explained in Section 6, we use the Meta-World `medium-replay` dataset from Choi et al. (2024), which is available in their official repository along with download instructions.

ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2022-NR071853 and RS-2023-00222663) and by AI-Bio Research Grant through Seoul National University.

REFERENCES

Gaon An, Junhyeok Lee, Xingdong Zuo, Norio Kosaka, Kyung-Min Kim, and Hyun Oh Song. Direct preference-based policy optimization without reward modeling. *Advances in Neural Information Processing Systems*, 36:70247–70266, 2023.

Philippe Artzner. Thinking coherently. *Risk*, 10:68–71, 1997.

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.

Mohak Bhardwaj, Tengyang Xie, Byron Boots, Nan Jiang, and Ching-An Cheng. Adversarial model for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

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.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International conference on machine learning*, pp. 783–792. PMLR, 2019.

Qi Cai, Zhuoran Yang, Chi Jin, and Zhaoran Wang. Provably efficient exploration in policy optimization. In *International Conference on Machine Learning*, pp. 1283–1294. PMLR, 2020.

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*, pp. 3773–3793. PMLR, 2022.

Yu Chen, Yihan Du, Pihe Hu, Siwei Wang, Desheng Wu, and Longbo Huang. Provably efficient iterated cvar reinforcement learning with function approximation and human feedback. In *The Twelfth International Conference on Learning Representations*, 2023.

Ching-An Cheng, Tengyang Xie, Nan Jiang, and Alekh Agarwal. Adversarially trained actor critic for offline reinforcement learning. In *International Conference on Machine Learning*, pp. 3852–3878. PMLR, 2022.

Pengyu Cheng, Yifan Yang, Jian Li, Yong Dai, Tianhao Hu, Peixin Cao, Nan Du, and Xiaolong Li. Adversarial preference optimization: Enhancing your alignment via rm-llm game. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 3705–3716, 2024.

Heewoong Choi, Sangwon Jung, Hongjoon Ahn, and Taesup Moon. Listwise reward estimation for offline preference-based reinforcement learning. In *Forty-first International Conference on Machine Learning*, 2024.

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. In *Forty-first International Conference on Machine Learning*, 2024.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

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

Joey Hejna and Dorsa Sadigh. Inverse preference learning: Preference-based rl without a reward function. *Advances in Neural Information Processing Systems*, 36, 2024.

Joey Hejna, Rafael Rafailov, Harshit Sikchi, Chelsea Finn, Scott Niekum, W. Bradley Knox, and Dorsa Sadigh. Contrastive preference learning: Learning from human feedback without reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Donald Joseph Hejna III and Dorsa Sadigh. Few-shot preference learning for human-in-the-loop rl. In *Conference on Robot Learning*, pp. 2014–2025. PMLR, 2023.

Minyoung Hwang, Gunmin Lee, Hogun Kee, Chan Woo Kim, Kyungjae Lee, and Songhwai Oh. Sequential preference ranking for efficient reinforcement learning from human feedback. *Advances in Neural Information Processing Systems*, 36, 2024.

Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. *Advances in neural information processing systems*, 31, 2018.

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

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*, pp. 651–673. PMLR, 2018.

Yachen Kang, Diyuan Shi, Jinxin Liu, Li He, and Donglin Wang. Beyond reward: Offline preference-guided policy optimization. In *International Conference on Machine Learning*, pp. 15753–15768. PMLR, 2023.

Changyeon Kim, Jongjin Park, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. Preference transformer: Modeling human preferences using transformers for RL. In *The Eleventh International Conference on Learning Representations*, 2023.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR)*, 2015. URL `https://arxiv.org/abs/1412.6980`. arXiv preprint arXiv:1412.6980.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

Kimin Lee, Laura M Smith, and Pieter Abbeel. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *International Conference on Machine Learning*, pp. 6152–6163. PMLR, 2021.

Anqi Li, Dipendra Misra, Andrey Kolobov, and Ching-An Cheng. Survival instinct in offline reinforcement learning. *Advances in neural information processing systems*, 36, 2024.

Xinran Liang, Katherine Shu, Kimin Lee, and Pieter Abbeel. Reward uncertainty for exploration in preference-based reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=OWZVD-l-ZrC.

Runze Liu, Fengshuo Bai, Yali Du, and Yaodong Yang. Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. *Advances in Neural Information Processing Systems*, 35:22270–22284, 2022.

James MacGlashan, Mark K Ho, Robert Loftin, Bei Peng, Guan Wang, David L Roberts, Matthew E Taylor, and Michael L Littman. Interactive learning from policy-dependent human feedback. In *International conference on machine learning*, pp. 2285–2294. PMLR, 2017.

Jacob Makar-Limanov, Arjun Prakash, Denizalp Goktas, Amy Greenwald, and Nora Ayanian. STA-RLHF: Stackelberg aligned reinforcement learning with human feedback. In *Coordination and Cooperation for Multi-Agent Reinforcement Learning Methods Workshop*, 2024.

Volodymyr Mnih. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

Ellen Novoseller, Yibing Wei, Yanan Sui, Yisong Yue, and Joel Burdick. Dueling posterior sampling for preference-based reinforcement learning. In *Conference on Uncertainty in Artificial Intelligence*, pp. 1029–1038. PMLR, 2020.

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.

Alizée Pace, Bernhard Schölkopf, Gunnar Ratsch, and Giorgia Ramponi. Preference elicitation for offline reinforcement learning. In *ICML 2024 Workshop: Foundations of Reinforcement Learning and Control – Connections and Perspectives*, 2024. URL https://openreview.net/forum?id=YGaRv4UCRh.

Jongjin Park, Younggyo Seo, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. SURF: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TfhfZLQ2EJO.

Barna Pásztor, Parnian Kassraie, and Andreas Krause. Bandits with preference feedback: A stackelberg game perspective. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=wIE991zhXH.

Aravind Rajeswaran, Igor Mordatch, and Vikash Kumar. A game theoretic framework for model based reinforcement learning. In *International conference on machine learning*, pp. 7953–7963. PMLR, 2020.

Marc Rigter, Bruno Lacerda, and Nick Hawes. Rambo-rl: Robust adversarial model-based offline reinforcement learning. *Advances in neural information processing systems*, 35:16082–16097, 2022.

Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. *Advances in Neural Information Processing Systems*, 26, 2013.

Aadirupa Saha, Aldo Pacchiano, and Jonathan Lee. Dueling rl: Reinforcement learning with trajectory preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 6263–6289. PMLR, 2023.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

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

Han Shen, Zhuoran Yang, and Tianyi Chen. Principled penalty-based methods for bilevel reinforcement learning and RLHF. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=Xb3IXEBYuw.

Daniel Shin, Anca Dragan, and Daniel S. Brown. Benchmarks and algorithms for offline preference-based reward learning. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Yuda Song, Yifei Zhou, Ayush Sekhari, Drew Bagnell, Akshay Krishnamurthy, and Wen Sun. Hybrid RL: Using both offline and online data can make RL efficient. In *The Eleventh International Conference on Learning Representations*, 2023.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Gokul Swamy, Christoph Dann, Rahul Kidambi, Steven Wu, and Alekh Agarwal. A minimaximalist approach to reinforcement learning from human feedback. In *Forty-first International Conference on Machine Learning*, 2024.

Masatoshi Uehara and Wen Sun. Pessimistic model-based offline reinforcement learning under partial coverage. In *International Conference on Learning Representations*, 2022.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Heinrich Von Stackelberg. *Market structure and equilibrium*. Springer Science & Business Media, 2010.

Garrett Warnell, Nicholas Waytowich, Vernon Lawhern, and Peter Stone. Deep tamer: Interactive agent shaping in high-dimensional state spaces. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136):1–46, 2017.

Runzhe Wu and Wen Sun. Making RL with preference-based feedback efficient via randomization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Pe2lo3QOvo.

Yichong Xu, Ruosong Wang, Lin Yang, Aarti Singh, and Artur Dubrawski. Preference-based reinforcement learning with finite-time guarantees. *Advances in Neural Information Processing Systems*, 33:18784–18794, 2020.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

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*, 2024a.

Wenhao Zhan, Masatoshi Uehara, Wen Sun, and Jason D. Lee. Provable reward-agnostic preference-based reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024b.

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*, pp. 43037–43067. PMLR, 2023.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Additional Related Work

Empirical PbRL. Incorporating preference feedback into reinforcement learning has been explored through various approaches. One standard method involves training a reward model from preferences, which is then used to train a standard RL algorithm (Christiano et al., 2017; Ibarz et al., 2018). A variety of techniques have emerged in this area, including unsupervised pre-training (Lee et al., 2021), exploration driven by uncertainty (Liang et al., 2022), data augmentation (Park et al., 2022), and meta-learning (Hejna III & Sadigh, 2023), to list a few. Another prominent line of research focuses on preference learning via active query methods (Shin et al., 2023; Hwang et al., 2024; Choi et al., 2024), which have shown strong empirical performance.

Beyond the conventional Markov reward model, some studies have proposed alternative reward structures. For example, Kim et al. (2023) employed transformer architectures for reward modeling, while Liu et al. (2022) and Hejna & Sadigh (2024) explored learning action-value functions rather than directly modeling rewards. Several approaches bypass explicit reward models altogether, instead optimizing policies directly (An et al., 2023; Kang et al., 2023; Hejna et al., 2024).

B Details on the Policy Evaluation Subroutine

We present a simple policy evaluation subroutine in Algorithm 3. It requires online rollouts and access to the reference policy. The idea of policy evaluation via online rollouts is adopted from Chang et al. (2024), while the analysis follows standard methods.

Algorithm 3 PE: Monte Carlo Policy Evaluation

Input: Reference policy μ, Current policy π^t, Estimated reward $\hat{r}$, Number of rollout K
for $h \in [H]$ **do**
  Collect K i.i.d. trajectories $\{(s_1^k, a_1^k, \ldots, s_H^k, a_H^k)\}_{k=1}^K$
  where $a_j^k \sim \mu_j(\cdot \mid s_j^k)$ for $j < h$, $a_h^k \sim \frac{1}{2}(\mu_h + \pi_h^t)(\cdot \mid s_h^k)$, and $a_j^k \sim \pi_j^t(\cdot \mid s_j^k)$ for $j > h$
  Compute $q_h^k = \sum_{j=h}^H \hat{r}(s_j^k, a_j^k)$, then set $\mathcal{D}_h^t = \{(s_h^k, a_h^k, q_h^k)\}_{k=1}^K$
  Least square value function estimation $\bar{Q}_h^t = \arg\min_{f \in \mathcal{F}} \frac{1}{K} \sum_{(s,a,q) \in \mathcal{D}_h^t} \left(f(s,a) - q\right)^2$
end for
Return $\{\bar{Q}_h^t\}_{h=1}^H$

We have the following guarantee.

Lemma B.1. *With probability at least $1 - \delta$, Algorithm 3 guarantees that, for every $(t, h) \in [T] \times [H]$,*

$$\mathbb{E}_{s \sim d_h^\mu, a \sim \frac{1}{2}(\mu^h + \pi_h^t)} \left[\left(\bar{Q}_h^t(s,a) - Q_{h,r^t}^{\pi^t}(s,a) \right)^2 \right] \leq \frac{c_3 R^2 \log(TH|\mathcal{F}|/\delta)}{K_2} =: \epsilon_{PE}^2$$

where c_3 is an absolute constant.

Proof. Since $\left\| Q_{h,r}^\pi \right\|_\infty \leq R$ for any policy π and $r \in \mathcal{R}^H$, Lemma E.4 with $B = R$ and $K = K_2$ leads to

$$\mathbb{E}_{s \sim d_h^\mu, a \sim \frac{1}{2}(\mu^h + \pi_h^t)} \left[\left(\bar{Q}_h^t(s,a) - Q_{h,r^t}^{\pi^t}(s,a) \right)^2 \right] \leq \frac{c_3 R^2 \log(|\mathcal{F}|/\delta)}{K_2}$$

for any fixed $(t, h) \in [T] \times [H]$. The union bound over all $(t, h) \in [T] \times [H]$ concludes the proof. □

C THEORETICAL ANALYSIS OF APPO-ROLLOUT

In this section, we provide theoretical analyses of APPO-rollout, a naïve algorithm for solving the optimization problem (2). The ideas presented in this section are relevant to the proof of Theorem 4.1, and the results are valuable for comparison with related works.

Before stating the theorem, we define step-wise concentrability, which is always bounded by C_{TR}.

Definition 2 (Step-wise concentrability)**.** $C_{ST} = \max_{h\in[H]} \sup_{(s,a)\in\mathcal{S}\times\mathcal{A}} \frac{d_h^{\pi^\star}(s,a)}{d_h^{\mu}(s,a)}$

Lemma C.1. *It always holds that $C_{ST} \leq C_{TR}$.*

Proof. For a fixed pair (s,a), consider the set of trajectories $\mathcal{T}(s,a) := \{\tau = (s_1, a_1, \ldots, s_H, a_H) : s_h = s, a_h = a\}$. Then we have that

$$d_h^{\pi}(s,a) = \int_{\mathcal{T}(s,a)} d^{\pi}(\tau) d\tau.$$

for any fixed policy π. Therefore, for every $(s,a) \in \mathcal{S}\times\mathcal{A}$, we have that

$$\frac{d_h^{\pi^\star}(s,a)}{d_h^{\mu}(s,a)} = \frac{\int_{\mathcal{T}(s,a)} d^{\pi^\star}(\tau)d\tau}{\int_{\mathcal{T}(s,a)} d^{\mu}(\tau)d\tau} \leq \sup_{\tau} \frac{d^{\pi^\star}(\tau)}{d^{\mu}(\tau)} = C_{\text{TR}}.$$

Taking the supremum on both sides concludes the proof. □

Theorem C.2. *Suppose Assumptions 1 and 4 hold. With probability at least $1-\delta$, Algorithm 1 with $\lambda = \Theta(C_{TR}), \lambda > C_{TR}, \eta = \sqrt{\frac{2\log|\mathcal{A}|}{R^2 T}}$ achieves*

$$\begin{aligned} &V_{1,r^\star}^{\pi^\star} - V_{1,r^\star}^{\hat{\pi}} \\ &\leq \mathcal{O}\left(\sqrt{\log\frac{|\mathcal{R}|}{\delta}}\left(\frac{C_{TR}\kappa\sqrt{H}}{\sqrt{M}} + \frac{R}{\sqrt{K_1}} + \frac{R}{\sqrt{N}}\right) + RH\sqrt{\frac{\log|\mathcal{A}|}{T}} + RH\sqrt{\frac{C_{ST}}{K_2}\log\frac{TH|\mathcal{F}|}{\delta}}\right). \end{aligned}$$

Setting $T = \Theta\left(\frac{R^2H^2\log|\mathcal{A}|}{\epsilon^2}\right)$, $N = K_1 = \Theta\left(\frac{R^2\log(|\mathcal{R}|/\delta)}{\epsilon^2}\right)$, $M = \Theta\left(\frac{C_{TR}^2\kappa^2 H\log(|\mathcal{R}|/\delta)}{\epsilon^2}\right)$, and $K_2 = \Theta\left(\frac{R^2H^2C_{ST}\log(TH|\mathcal{F}|/\delta)}{\epsilon^2}\right)$, Algorithm 2 achieves ϵ-optimal policy, i.e. $V_{1,r^\star}^{\pi^\star} - V_{1,r^\star}^{\bar{\pi}} \leq \epsilon$.

Discussion on Theorem C.2. We compare this bound with PbRL algorithms that assume a known transition model or allow online rollouts. In comparison to FREEHAND (Zhan et al., 2024a), APPO-rollout achieves a nearly identical rate for labeled data, but unlike FREEHAND, it requires additional unlabeled trajectories. This represents a trade-off between statistical efficiency and computational complexity, as FREEHAND relies on solving a nearly intractable nested optimization problem. Another comparable algorithm is DR-PO (Chang et al., 2024), which establishes a sample complexity of $\Theta\left(\frac{(C_{\text{TR}}+C_{\text{SFT}})\kappa^2\log(|\mathcal{R}|/\delta)}{\epsilon^2}\right)$ for labeled data. Unlike APPO-rollout, DR-PO assumes homogeneous rewards, which removes dependence on H from the bound. While their bound is tighter in C_{TR}, it comes at the cost of dependence on an additional factor, $C_{\text{SFT}} = \sup_{\pi\in D}\sup_{\tau}\frac{d^{\pi}(\tau)}{d^{\mu}(\tau)}$, where D is a set of policies close to μ in terms of KL divergence. This additional term arises because DR-PO does not explicitly ensure conservatism.

For simplicity, we introduce some notation regarding optimization objectives in Algorithm 1. For $r, \tilde{r} \in \mathcal{R}^H$, we define

$$\hat{\mathcal{L}}_{\text{opt}}^t(r;\tilde{r}) := \mathbb{E}_{\tau\sim\mathcal{D}_{\text{rollout}}}[r(\tau)] - \mathbb{E}_{\tau\sim\mathcal{D}_{\text{traj}}}[r(\tau)] + \lambda\hat{\mathcal{E}}_{\mathcal{D}_{\text{traj}}}(r;\tilde{r})$$

and its population version as

$$\mathcal{L}_{\text{opt}}^t(r;\tilde{r}) := \mathbb{E}_{\tau\sim\pi^t}[r(\tau)] - \mathbb{E}_{\tau\sim\mu}[r(\tau)] + \lambda\mathcal{E}(r;\tilde{r}).$$

C.1 Optimization Error

In this section, we prove that the (finite-sample) optimization objective $\hat{\mathcal{L}}^t_{\text{opt}}(r;\hat{r})$ is close to its population version, $\mathcal{L}^t_{\text{opt}}(r;\tilde{r})$. The result ensures that r^t is a good approximation of the solution to the optimization program with infinite samples, i.e.

$$r^t \approx \arg\min_{r\in\mathcal{R}^H} \mathcal{L}^t_{\text{opt}}(r;\hat{r}).$$

Lemma C.3. *With probability at least $1-\delta/2$, for all $t\in[T]$, we have*

$$\mathcal{L}_{opt}(r^t;\hat{r}) \le \mathcal{L}_{opt}(r^\star;\hat{r}) + 2\tilde{\epsilon}_{approx}$$

where $\tilde{\epsilon}_{approx}$ is defined in Lemma C.4.

Proof. We have the following decomposition:

$$\begin{aligned}
&\mathcal{L}^t_{\text{opt}}(r^t;\hat{r}) - \mathcal{L}^t_{\text{opt}}(r^\star;\hat{r}) \\
&= \underbrace{\mathcal{L}^t_{\text{opt}}(r^t;\pi^t) - \hat{\mathcal{L}}^t_{\text{opt}}(r^t;\hat{r})}_{\text{(I)}} + \underbrace{\hat{\mathcal{L}}^t_{\text{opt}}(r^t;\hat{r}) - \hat{\mathcal{L}}^t_{\text{opt}}(r^\star;\hat{r})}_{\text{(II)}} + \underbrace{\hat{\mathcal{L}}^t_{\text{opt}}(r^\star;\hat{r}) - \mathcal{L}^t_{\text{opt}}(r^\star;\hat{r})}_{\text{(III)}}
\end{aligned}$$

Conditioned on the event defined by Lemma D.2, (I) and (III) are bounded by ϵ_{opt}. Moreover, the optimality of r^t implies (II)≤ 0. □

Lemma C.4. *With probability at least $1-\delta/2$, for every $t\in[T]$ and $r\in\mathcal{R}^H$, it holds that*

$$\left|\mathcal{L}^t_{opt}(r;\hat{r}) - \hat{\mathcal{L}}^t_{opt}(r;\hat{r})\right| \le R\sqrt{\frac{\log(6|\mathcal{R}|/\delta)}{2K_1}} + 2R\sqrt{\frac{2\log(6|\mathcal{R}|/\delta)}{N}} := \tilde{\epsilon}_{approx}$$

Proof. Fix $r\in\mathcal{R}^H$, and note that

$$\begin{aligned}
&\left|\mathcal{L}^t_{\text{opt}}(r;\hat{r}) - \hat{\mathcal{L}}^t_{\text{opt}}(r;\hat{r})\right| \\
&\le \left|\mathbb{E}_{\tau\sim\mathcal{D}^t_{\text{rollout}}}[r(\tau)] - \mathbb{E}_{\tau\sim\pi^t}[r(\tau)]\right| + \left|\mathbb{E}_{\tau\sim\mathcal{D}_{\text{traj}}}[r(\tau)] - \mathbb{E}_{\tau\sim\mu}[r(\tau)]\right| \\
&\quad + \left|\mathbb{E}_{(\tau^0,\tau^1)\sim\mathcal{D}_{\text{traj}}}\left[(r-\hat{r})(\tau^0) - (r-\hat{r})(\tau^1)\right] - \mathbb{E}_{(\tau^0,\tau^1)\sim\mu}\left[(r-\hat{r})(\tau^0) - (r-\hat{r})(\tau^1)\right]\right|.
\end{aligned}$$

Since $|r(\tau)|\le R$ and $|(r-\hat{r})(\tau)|\le R$ for any trajectory τ, each term can be bounded by Hoeffding inequality. Specifically, each of these three events occurs with probability at least $1-\delta/6$:

$$\begin{aligned}
&\left|\mathbb{E}_{\tau\sim\mathcal{D}^t_{\text{rollout}}}[r(\tau)] - \mathbb{E}_{\tau\sim\pi^t}[r(\tau)]\right| \le R\sqrt{\frac{\log(6/\delta)}{2K_1}}, \\
&\left|\mathbb{E}_{\tau\sim\mathcal{D}_{\text{traj}}}[r(\tau)] - \mathbb{E}_{\tau\sim\mu}[r(\tau)]\right| \le R\sqrt{\frac{\log(6/\delta)}{2N}}, \\
&\left|\mathbb{E}_{(\tau^0,\tau^1)\sim\mathcal{D}_{\text{traj}}}\left[(r-\hat{r})(\tau^0) - (r-\hat{r})(\tau^1)\right] - \mathbb{E}_{(\tau^0,\tau^1)\sim\mu}\left[(r-\hat{r})(\tau^0) - (r-\hat{r})(\tau^1)\right]\right| \le 2R\sqrt{\frac{\log(6/\delta)}{2N}}.
\end{aligned}$$

Taking union bound over these events and all $r\in\mathcal{R}^H$, with probability at least $1-\delta/2$, it holds that

$$\begin{aligned}
\left|\mathcal{L}^t_{\text{opt}}(r;\hat{r}) - \hat{\mathcal{L}}^t_{\text{opt}}(r;\hat{r})\right| &\le R\sqrt{\frac{\log(6|\mathcal{R}|/\delta)}{2K_1}} + R\sqrt{\frac{\log(6|\mathcal{R}|/\delta)}{2N}} + 2R\sqrt{\frac{\log(6|\mathcal{R}|/\delta)}{2N}} \\
&\le R\sqrt{\frac{\log(6|\mathcal{R}|/\delta)}{2K_1}} + 2R\sqrt{\frac{2\log(6|\mathcal{R}|/\delta)}{N}}
\end{aligned}$$

for every $r\in\mathcal{R}^H$. □

C.2 POLICY UPDATE

We present the guarantee regarding the policy update steps. The proofs in this section are based on the standard analysis of the natural policy gradient (also referred to as trust region policy optimization) (Cai et al., 2020; Chang et al., 2024).

Lemma C.5. *With probability at least* $1-\delta/4$*, it holds that*

$$\frac{1}{T}\sum_{t=1}^{T}\left(V_{1,r^t}^{\pi^\star}(s_1)-V_{1,r^t}^{\pi^t}(s_1)\right)\leq RH\sqrt{\frac{\log|\mathcal{A}|}{2T}}+2H\epsilon_{PE}\sqrt{2C_{ST}}$$

Proof of Lemma C.5. The performance difference lemma (Lemma E.1) implies that

$$\begin{aligned}
&\sum_{t=1}^{T}\left(V_{1,r^t}^{\pi^\star}(s_1)-V_{1,r^t}^{\pi^t}(s_1)\right)\\
&=\sum_{t=1}^{T}\mathbb{E}_{\pi^\star}\left[\sum_{h=1}^{H}\langle Q_{h,r^t}^{\pi^t}(s_h,\cdot),\pi_h^\star(\cdot\mid s_h)-\pi_h^t(\cdot\mid s_h)\rangle\right]\\
&=\underbrace{\sum_{t=1}^{T}\sum_{h=1}^{H}\mathbb{E}_{s\sim d_h^{\pi^\star}}\left[\langle\bar{Q}_h^t(s,\cdot),\pi_h^\star(\cdot\mid s)-\pi_h^t(\cdot\mid s)\rangle\right]}_{\text{(I)}}\\
&\quad+\underbrace{\sum_{t=1}^{T}\sum_{h=1}^{H}\mathbb{E}_{s\sim d_h^{\pi^\star}}\left[\langle(Q_{h,r^t}^{\pi^t}-\bar{Q}_h^t)(s,\cdot),\pi_h^\star(\cdot\mid s_h)-\pi_h^t(\cdot\mid s_h)\rangle\right]}_{\text{(II)}}
\end{aligned}$$

Bounding (I). Decompose the inner product inside the expectation:

$$\begin{aligned}
&\langle\eta\bar{Q}_h^t(s_h,\cdot),\pi_h^\star(\cdot\mid s)-\pi_h^t(\cdot\mid s)\rangle\\
&\langle\eta\bar{Q}_h^t(s_h,\cdot),\pi_h^\star(\cdot\mid s)-\pi_h^{t+1}(\cdot\mid s)\rangle+\langle\eta\bar{Q}_h^t(s_h,\cdot),\pi_h^{t+1}(\cdot\mid s)-\pi_h^t(\cdot\mid s)\rangle\\
&\leq\langle\eta\bar{Q}_h^t(s_h,\cdot),\pi_h^\star(\cdot\mid s)-\pi_h^{t+1}(\cdot\mid s)\rangle+\eta\left\|\bar{Q}_h^t(s_h,\cdot)\right\|_\infty\left\|\pi_h^\star(\cdot\mid s)-\pi_h^{t+1}(\cdot\mid s)\right\|_1\\
&\leq\langle\eta\bar{Q}_h^t(s_h,\cdot),\pi_h^\star(\cdot\mid s)-\pi_h^{t+1}(\cdot\mid s)\rangle+\eta R\left\|\pi_h^\star(\cdot\mid s)-\pi_h^{t+1}(\cdot\mid s)\right\|_1
\end{aligned}\tag{9}$$

where we use Hölder's inequality with the fact that $\left\|\bar{Q}_h^t\right\|_\infty\leq R$. Now recall that the policy update step (Line 7) in Algorithm 1 leads to

$$\pi_h^{t+1}(\cdot\mid s)=\frac{1}{Z_h^t(s)}\pi_h^t(\cdot\mid s)\exp\left(\eta\bar{Q}_h^t(s,\cdot)\right)$$

where $Z_h^t(s)=\sum_{a\in\mathcal{A}}\pi_h^t(a\mid s)\exp\left(\eta\bar{Q}_h^t(s,a)\right)$. Using the relationship $\eta\bar{Q}_h^t(s,a)=\log Z_h^t(s)+\log\pi_h^{t+1}(a\mid s)-\log\pi_h^t(a\mid s)$, it holds that

$$\begin{aligned}
&\langle\eta\bar{Q}_h^t(s_h,\cdot),\pi_h^\star(\cdot\mid s)-\pi_h^{t+1}(\cdot\mid s)\rangle\\
&=\langle\log Z_h^t(s)+\log\pi_h^{t+1}(\cdot\mid s)-\log\pi_h^t(\cdot\mid s),\pi_h^\star(\cdot\mid s)-\pi_h^{t+1}(\cdot\mid s)\rangle\\
&=\langle\log\pi_h^{t+1}(\cdot\mid s)-\log\pi_h^t(\cdot\mid s),\pi^\star(\cdot\mid s)-\pi_h^{t+1}(\cdot\mid s)\rangle\\
&=\langle\log\pi_h^{t+1}(\cdot\mid s)-\log\pi_h^t(\cdot\mid s),\pi^\star(\cdot\mid s)\rangle-D_{KL}\left(\pi_h^{t+1}(\cdot\mid s)\|\pi_h^t(\cdot\mid s)\right)\\
&=\langle\log\frac{\pi_h^{t+1}(\cdot\mid s)}{\pi_h^\star(\cdot\mid s)}+\log\frac{\pi_h^\star(\cdot\mid s)}{\pi_h^t(\cdot\mid s)},\pi_h^\star(\cdot\mid s)\rangle-D_{KL}\left(\pi_h^{t+1}(\cdot\mid s)\|\pi_h^t(\cdot\mid s)\right)\\
&=D_{KL}\left(\pi_h^\star(\cdot\mid s)\|\pi_h^t(\cdot\mid s)\right)-D_{KL}\left(\pi_h^\star(\cdot\mid s)\|\pi_h^{t+1}(\cdot\mid s)\right)-D_{KL}\left(\pi_h^{t+1}(\cdot\mid s)\|\pi_h^t(\cdot\mid s)\right)\\
&\leq D_{KL}\left(\pi_h^\star(\cdot\mid s)\|\pi_h^t(\cdot\mid s)\right)-D_{KL}\left(\pi_h^\star(\cdot\mid s)\|\pi_h^{t+1}(\cdot\mid s)\right)-\frac{1}{2}\left\|\pi_h^\star(\cdot\mid s)-\pi_h^{t+1}(\cdot\mid s)\right\|_1^2
\end{aligned}$$

where the second equality holds since $Z_h^t(s)$ is a constant given s, and the last inequality holds due to Pinsker's inequality. Combining this bound with (9), we obtain

$$
\begin{aligned}
&\sum_{t=1}^{T} \langle \eta \bar{Q}_h^t(s_h, \cdot), \pi_h^\star(\cdot \mid s) - \pi_h^t(\cdot \mid s) \rangle \\
&= \sum_{t=1}^{T} \left(D_{KL}\left(\pi_h^\star(\cdot \mid s) \| \pi_h^t(\cdot \mid s)\right) - D_{KL}\left(\pi_h^\star(\cdot \mid s) \| \pi_h^{t+1}(\cdot \mid s)\right)\right) \\
&\quad + \sum_{t=1}^{T} \left(\eta R \left\| \pi_h^\star(\cdot \mid s) - \pi_h^{t+1}(\cdot \mid s) \right\|_1 - \frac{1}{2} \left\| \pi_h^\star(\cdot \mid s) - \pi_h^{t+1}(\cdot \mid s) \right\|_1^2 \right) \\
&\leq \sum_{t=1}^{T} \left(D_{KL}\left(\pi_h^\star(\cdot \mid s) \| \pi_h^t(\cdot \mid s)\right) - D_{KL}\left(\pi_h^\star(\cdot \mid s) \| \pi_h^{t+1}(\cdot \mid s)\right)\right) + \sum_{t=1}^{T} \frac{\eta^2 R^2}{2} \\
&= D_{KL}\left(\pi_h^\star(\cdot \mid s) \| \pi_h^1(\cdot \mid s)\right) - D_{KL}\left(\pi_h^\star(\cdot \mid s) \| \pi_h^{T+1}(\cdot \mid s)\right) + \frac{\eta^2 R^2 T}{2} \\
&\leq \log |\mathcal{A}| + \frac{\eta^2 R^2 T}{2}
\end{aligned}
$$

where the first inequality holds since $\forall x \in \mathbb{R}\, ax - x^2/2 \leq a^2/2$, and the second inequality holds due to the fact that $\pi_h^1 = \text{Unif}(\mathcal{A})$. Finally, setting $\eta = \sqrt{\frac{2 \log |\mathcal{A}|}{R^2 T}}$, (I) is bounded by

$$
\begin{aligned}
\text{(I)} &= \sum_{h=1}^{H} \mathbb{E}_{s \sim d_h^{\pi^\star}} \left[\sum_{t=1}^{T} \langle \bar{Q}_h^t(s, \cdot), \pi^\star(\cdot \mid s) - \pi^t(\cdot \mid s) \rangle \right] \\
&\leq \sum_{h=1}^{H} \frac{\log |\mathcal{A}|}{\eta} + \frac{\eta R^2 T}{2} = RH \sqrt{\frac{T \log |\mathcal{A}|}{2}}
\end{aligned}
$$

Bounding (II). We condition on the event defined by Lemma B.1. Then we have

$$
\begin{aligned}
&\left| \mathbb{E}_{s \sim d_h^{\pi^\star}} \left[\langle (Q_{h,r^t}^{\pi^t} - \bar{Q}_h^t)(s, \cdot), \pi_h^\star \rangle \right] \right| \\
&= \left| \mathbb{E}_{(s,a) \sim d_h^{\pi^\star}} \left[Q_{h,r^t}^{\pi^t}(s,a) - \bar{Q}_h^t(s,a) \right] \right| \\
&\leq \sqrt{\mathbb{E}_{(s,a) \sim d_h^{\pi^\star}} \left[\left(Q_{h,r^t}^{\pi^t}(s,a) - \bar{Q}_h^t(s,a) \right)^2 \right]} \\
&\leq \sqrt{2 \left(\max_{h \in [H]} \sup_{(s,a) \in \mathcal{S} \times \mathcal{A}} \frac{d_h^{\pi^\star}(s,a)}{d_h^{\mu}(s,a)} \right) \mathbb{E}_{s \sim d_h^{\mu}, a \sim \frac{1}{2}(\pi_h^t + \mu_h)} \left[\left(Q_{h,r^t}^{\pi^t}(s,a) - \bar{Q}_h^t(s,a) \right)^2 \right]} \\
&\leq \sqrt{2 C_{\text{ST}} \epsilon_{\text{PE}}^2}
\end{aligned}
$$

where the first inequality holds due to Jensen's inequality, the second inequality uses importance sampling, and the last inequality uses Lemma B.1.

$$
\begin{aligned}
&\left|\mathbb{E}_{s\sim d_h^{\pi^\star}}\left[\langle (Q_{h,r^t}^{\pi^t} - \bar{Q}_h^t)(s,\cdot), \pi_h^t\rangle\right]\right| \\
&= \left|\mathbb{E}_{s\sim d_h^{\pi^\star}, a\sim\pi_h^t}\left[Q_{h,r^t}^{\pi^t}(s,a) - \bar{Q}_h^t(s,a)\right]\right| \\
&\leq \sqrt{\mathbb{E}_{s\sim d_h^{\pi^\star}, a\sim\pi_h^t}\left[\left(Q_{h,r^t}^{\pi^t}(s,a) - \bar{Q}_h^t(s,a)\right)^2\right]} \\
&\leq \sqrt{2\left(\max_{h\in[H]}\sup_{s\in\mathcal{S}}\frac{d_h^{\pi^\star}(s)}{d_h^{\mu}(s)}\right)\mathbb{E}_{s\sim d_h^{\mu}, a\sim\frac{1}{2}(\pi_h^t+\mu_h)}\left[\left(Q_{h,r^t}^{\pi^t}(s,a) - \bar{Q}_h^t(s,a)\right)^2\right]} \\
&\leq \sqrt{2\left(\max_{h\in[H]}\sup_{s\in\mathcal{S}}\frac{d_h^{\pi^\star}(s)}{d_h^{\mu}(s)}\right)\mathbb{E}_{s\sim d_h^{\mu}, a\sim\frac{1}{2}(\pi_h^t+\mu_h)}\left[\left(Q_{h,r^t}^{\pi^t}(s,a) - \bar{Q}_h^t(s,a)\right)^2\right]} \\
&\leq \sqrt{2C_{\text{ST}}\epsilon_{\text{PE}}^2}.
\end{aligned}
$$

Therefore, we obtain the bound

$$
\begin{aligned}
\text{(II)} &\leq \sum_{t=1}^{T}\sum_{h=1}^{H}\left|\mathbb{E}_{s\sim d_h^{\pi^\star}}\left[\langle (Q_{h,r^t}^{\pi^t} - \bar{Q}_h^t)(s,\cdot), \pi_h^\star(\cdot \mid s_h)\rangle\right]\right| \\
&\quad + \sum_{t=1}^{T}\sum_{h=1}^{H}\left|\mathbb{E}_{s\sim d_h^{\pi^\star}}\left[\langle (Q_{h,r^t}^{\pi^t} - \bar{Q}_h^t)(s,\cdot), \pi_h^t(\cdot \mid s_h)\rangle\right]\right| \\
&\leq 2TH\epsilon_{\text{PE}}\sqrt{2C_{\text{ST}}}.
\end{aligned}
$$

We conclude the proof by combining the bounds on (I) and (II). □

Now we prove Theorem C.2 based on the lemmas.

Proof of Theorem C.2. We condition on the event defined by Lemma E.2 (with $\delta' = \delta/4$), Lemma C.3, and Lemma C.5, which hold simultaneously with probability at least $1-\delta$. Consider the following sub-optimality decomposition at step t:

$$
\begin{aligned}
V_{1,r^\star}^{\pi^\star} - V_{1,r^\star}^{\pi^t} &= V_{1,r^\star}^{\pi^\star} - V_{1,\hat{r}}^{\pi^\star} + V_{1,\hat{r}}^{\pi^\star} - V_{1,r^t}^{\pi^\star} + V_{1,r^t}^{\pi^\star} - V_{1,r^\star}^{\pi^t} + V_{1,r^t}^{\pi^t} - V_{1,r^t}^{\pi^t} \\
&= \underbrace{V_{1,r^\star-\hat{r}}^{\pi^\star} - V_{1,r^\star-\hat{r}}^{\mu}}_{\text{(I) : MLE estimation error}} \\
&\quad + \underbrace{V_{1,\hat{r}-r^t}^{\pi^\star} - V_{1,\hat{r}-r^t}^{\mu} - V_{1,r^\star}^{\pi^t} + V_{1,r^\star}^{\mu} + V_{1,r^t}^{\pi^t} - V_{1,r^t}^{\mu}}_{\text{(II) : Optimization error}} \\
&\quad + \underbrace{V_{1,r^t}^{\pi^\star} - V_{1,r^t}^{\pi^t}}_{\text{(III) : Policy update regret}},
\end{aligned} \tag{10}
$$

where we omit the initial state s_1 for simplicity.

Bounding (I). Since we condition on the event defined by Lemma E.2, we have

$$
\begin{aligned}
\text{(I)} &= V_{1,r^\star-\hat{r}}^{\pi^\star} - V_{1,r^\star-\hat{r}}^{\mu} \\
&= \mathbb{E}_{\tau^0\sim\pi^\star,\tau^1\sim\mu}\left[r^\star(\tau^0) - r^\star(\tau^1) - \hat{r}(\tau^0) + \hat{r}(\tau^1)\right] \\
&\leq \sqrt{\mathbb{E}_{\tau^0\sim\pi^\star,\tau^1\sim\mu}\left[|r^\star(\tau^0) - r^\star(\tau^1) - \hat{r}(\tau^0) + \hat{r}(\tau^1)|^2\right]} \\
&\leq \sqrt{C_{\text{TR}}\mathbb{E}_{\tau^0,\tau^1\sim\mu}\left[|r^\star(\tau^0) - r^\star(\tau^1) - \hat{r}(\tau^0) + \hat{r}(\tau^1)|^2\right]} \\
&\leq \sqrt{C_{\text{TR}}}\epsilon_r(\delta/4).
\end{aligned}
$$

Bounding (II). We can relate the terms $V^{\pi^\star}_{1,\hat{r}-r^t} - V^{\mu}_{1,\hat{r}-r^t}$ to $\mathcal{E}(r^t; P^\star, \hat{r})$. By Assumption 4, we have that

$$\begin{aligned}
&V^{\pi^\star}_{1,\hat{r}-r^t} - V^{\mu}_{1,\hat{r}-r^t} \\
&= \mathbb{E}_{\tau^0\sim\pi^\star,\tau^1\sim\mu}\left[\hat{r}(\tau^0) - \hat{r}(\tau^1) - r^t(\tau^0) + r^t(\tau^1)\right] \\
&\leq C_{\text{TR}}\mathbb{E}_{\tau^0,\tau^1\sim\mu}\left[|\hat{r}(\tau^0) - \hat{r}(\tau^1) - r^t(\tau^0) + r^t(\tau^1)|\right] \\
&= C_{\text{TR}}\mathcal{E}(r^t;\hat{r}) \leq \lambda\mathcal{E}(r^t;\hat{r})
\end{aligned}$$

where the last inequality holds since $\mathcal{E}(r^t;\hat{r})$ is non-negative and $\lambda \geq C_{\text{TR}}$. Further, Lemma C.3 implies

$$\begin{aligned}
\lambda\mathcal{E}(r^t;\hat{r}) &\leq V^{\pi^t}_{1,r^\star} - V^{\mu}_{1,r^\star} - V^{\pi^t}_{1,r^t} + V^{\mu}_{1,r^t} + \lambda\mathcal{E}(r^\star;\hat{r}) + 2\tilde{\epsilon}_{approx} \\
&\leq V^{\pi^t}_{1,r^\star} - V^{\mu}_{1,r^\star} - V^{\pi^t}_{1,r^t} + V^{\mu}_{1,r^t} + \lambda\tilde{\epsilon}_r(\delta/4) + 2\tilde{\epsilon}_{approx}
\end{aligned}$$

where the last inequality holds due to Lemma E.2:

$$\begin{aligned}
\mathcal{E}(r^\star;\hat{r}) &= \mathbb{E}_{\tau^0,\tau^1\sim\mu}\left[|\hat{r}(\tau^0) - \hat{r}(\tau^1) - r^\star(\tau^0) + r^\star(\tau^1)|\right] \\
&\leq \sqrt{\mathbb{E}_{\tau^0,\tau^1\sim\mu}\left[|\hat{r}(\tau^0) - \hat{r}(\tau^1) - r^\star(\tau^0) + r^\star(\tau^1)|^2\right]} \leq \epsilon_r(\delta/4).
\end{aligned}$$

Therefore, we have

$$\text{(II)} \leq \lambda\epsilon_r(\delta/4) + 2\tilde{\epsilon}_{approx}.$$

Bounding Sub-optimality. Putting the bounds on (I) and (II) into (10), we have

$$\begin{aligned}
&V^{\pi^\star}_{1,r^\star} - V^{\pi^t}_{1,r^\star} \\
&\leq \sqrt{C_{\text{TR}}}\epsilon_r(\delta/4) + \lambda\epsilon_r(\delta/4) + 2\tilde{\epsilon}_{approx} + V^{\pi^\star}_{1,r^t} - V^{\pi^t}_{1,r^t}
\end{aligned} \tag{11}$$

Since Algorithm 1 returns the mixture policy $\hat{\pi} = \frac{1}{T}\sum_{t=1}^{T}\pi^t$, the sub-optimality is $V^{\pi^\star}_{1,r^\star} - V^{\hat{\pi}}_{1,r^\star} = \frac{1}{T}\sum_{t=1}^{T}\left(V^{\pi^\star}_{1,r^\star} - V^{\pi^t}_{1,r^\star}\right)$. Using the bound in (11) and Lemma C.5, it holds that

$$\begin{aligned}
&V^{\pi^\star}_{1,r^\star} - V^{\hat{\pi}}_{1,r^\star} \\
&= \frac{1}{T}\sum_{t=1}^{T}\left(V^{\pi^\star}_{1,r^\star} - V^{\pi^t}_{1,r^\star}\right) \\
&\leq \sqrt{C_{\text{TR}}}\epsilon_r(\delta/4) + \lambda\epsilon_r(\delta/4) + 2\tilde{\epsilon}_{approx} + \frac{1}{T}\sum_{t=1}^{T}\left(V^{\pi^\star}_{1,r^t} - V^{\pi^t}_{1,r^t}\right) \\
&\leq \sqrt{C_{\text{TR}}}\epsilon_r(\delta/4) + \lambda\epsilon_r(\delta/4) + 2\tilde{\epsilon}_{approx} + RH\sqrt{\frac{\log|\mathcal{A}|}{2T}} + 2H\epsilon_{\text{PE}}\sqrt{C_{\text{ST}}} \\
&\leq \mathcal{O}\left(\sqrt{\log\frac{|\mathcal{R}|}{\delta}}\left(\frac{C_{\text{TR}}\kappa\sqrt{H}}{\sqrt{M}} + \frac{R}{\sqrt{K_1}} + \frac{R}{\sqrt{N}}\right) + RH\sqrt{\frac{\log|\mathcal{A}|}{T}} + RH\sqrt{\frac{C_{\text{ST}}}{K_2}\log\frac{TH|\mathcal{F}|}{\delta}}\right)
\end{aligned}$$

□

D Detailed Proof of Theorem 4.1

For simplicity, we introduce some notations regarding optimization objectives in Algorithm 2. For $f \in \mathcal{F}^H$, we define

$$\hat{\mathcal{L}}^t_{\text{opt}}(f; \tilde{P}, \tilde{r}) := \sum_{h=1}^{H} \mathbb{E}_{(s_h,a_h)\sim\mathcal{D}_{\text{traj}}} \left[f_h \circ \pi^t_h(s_h) - f_h(s_h, a_h) \right] + \lambda \hat{\mathcal{E}}_{\mathcal{D}_{\text{traj}}}(f; \tilde{P}, \tilde{r})$$

and its population version as

$$\mathcal{L}^t_{\text{opt}}(f; \tilde{P}, \tilde{r}) := \sum_{h=1}^{H} \mathbb{E}_{(s_h,a_h)\sim d^\mu_h} \left[f_h \circ \pi^t_h(s_h) - f_h(s_h, a_h) \right] + \lambda \mathcal{E}(f; \tilde{P}, \tilde{r})$$

D.1 Optimization Error

In this section, we prove that the (finite-sample) optimization objective $\hat{\mathcal{L}}^t_{\text{opt}}(f; \hat{P}, \hat{r})$ is close to its population version $\mathcal{L}^t_{\text{opt}}(f; \tilde{P}, \tilde{r})$. The result ensures that f^t is a good approximation for the solutions to the optimization program with infinite samples, i.e.

$$f^t \approx \arg\min_{f \in \mathcal{F}^H} \mathcal{L}^t_{\text{opt}}(f; P^\star, \hat{r}).$$

Remark. For this section, we assume that the maximum likelihood transition estimator $\hat{P}$ is computed using half of $\mathcal{D}_{\text{traj}}$, and the losses $\hat{\mathcal{L}}^t_{\text{opt}}(f; \hat{P}, \hat{r})$ are computed from the other half. This increases the sample complexity only by a constant factor but helps avoid union bound over $\mathcal{P}$ in the proof of Lemma D.2.

Lemma D.1. *With probability at least $1 - \delta/2$, for all $t \in [T]$, we have that*

$$\mathcal{L}^t_{opt}(f^t; \hat{r}) \le \mathcal{L}^t_{opt}(Q^{\pi^t}; \hat{r}) + 2\epsilon_{approx}$$

where ϵ_{approx} is defined in Lemma D.2.

Proof. Consider this decomposition:

$$\begin{aligned}
&\mathcal{L}^t_{\text{opt}}(f^t; \hat{r}) - \mathcal{L}^t_{\text{opt}}(Q^{\pi^t}; \hat{r}) \\
&= \underbrace{\mathcal{L}^t_{\text{opt}}(f^t; \hat{r}) - \hat{\mathcal{L}}^t_{\text{opt}}(f^t; \hat{P}, \hat{r})}_{\text{(I)}} + \underbrace{\hat{\mathcal{L}}^t_{\text{opt}}(f^t; \hat{P}, \hat{r}) - \hat{\mathcal{L}}^t_{\text{opt}}(Q^{\pi^t}; \hat{P}, \hat{r})}_{\text{(II)}} + \underbrace{\hat{\mathcal{L}}^t_{\text{opt}}(Q^{\pi^t}; \hat{P}, \hat{r}) - \mathcal{L}^t_{\text{opt}}(Q^{\pi^t}; \hat{r})}_{\text{(III)}}.
\end{aligned}$$

Conditioned on the event defined by Lemma D.2, (I) and (III) are bounded by ϵ_{approx}. Moreover, the optimality of f^t implies (II)≤ 0. □

Lemma D.2. *With probability at least $1 - \delta/2$, for every $t \in [T]$ and $f \in \mathcal{F}^H$, it holds that*

$$\left| \hat{\mathcal{L}}^t_{opt}(f; \hat{P}, \hat{r}) - \mathcal{L}^t_{opt}(f; \hat{r}) \right| \le 8R\sqrt{\frac{H^3 T \log(8H|\mathcal{F}|/\delta)}{N}} + 2RH\epsilon_P(\delta/8) := \epsilon_{approx}.$$

Proof. Due to the policy update in Line 7 of Algorithm 2, the policies $\{\pi^t_h\}_{(t,h)\in[T]\times[H]}$ belongs to the following function class:

$$\Pi = \left\{ \pi(a \mid s) = \frac{\exp\left(\sum_{i=1}^T \eta f^i(s,a)\right)}{\sum_{a'\in\mathcal{A}} \exp\left(\sum_{i=1}^T \eta f^i(s,a')\right)} : f^i \in \mathcal{F} \text{ for all } i \in [T] \right\}.$$

It is clear that $|\Pi| \le |\mathcal{F}|^T$.

Step 1. Fix $h \in [H]$, $f \in \mathcal{F}$, and $\pi \in \Pi$. Since $|f \circ \pi(s)| \leq R$ for all $s \in \mathcal{S}$, Hoeffding inequality implies that

$$\left|\mathbb{E}_{s_h \in \mathcal{D}_{\text{traj}}}\left[f \circ \pi(s_h)\right] - \mathbb{E}_{s_h \sim d_h^\mu}\left[f \circ \pi(s_h)\right]\right| \leq R\sqrt{\frac{\log(8/\delta)}{2N}}$$

with probability at least $1 - \delta/8$. Similarly, since $|f(s,a)| \leq R$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, it holds that

$$\left|\mathbb{E}_{(s_h,a_h) \sim \mathcal{D}_{\text{traj}}}\left[f(s_h,a_h)\right] - \mathbb{E}_{(s_h,a_h) \sim d_h^\mu}\left[f(s_h,a_h)\right]\right| \leq R\sqrt{\frac{\log(8/\delta)}{2N}}$$

with probability at least $1 - \delta/8$. Thus, with probability at least $1 - \delta/4$, we have

$$\begin{aligned}
&\left|\mathbb{E}_{(s_h,a_h) \sim \mathcal{D}_{\text{traj}}}\left[f \circ \pi(s_h) - f(s_h,a_h)\right] - \mathbb{E}_{(s_h,a_h) \sim d_h^\mu}\left[f \circ \pi(s_h) - f(s_h,a_h)\right]\right| \\
&\leq \left|\mathbb{E}_{(s_h,a_h) \sim \mathcal{D}_{\text{traj}}}\left[f \circ \pi(s_h)\right] - \mathbb{E}_{(s_h,a_h) \sim d_h^\mu}\left[f \circ \pi(s_h)\right]\right| \\
&\quad + \left|\mathbb{E}_{(s_h,a_h) \sim \mathcal{D}_{\text{traj}}}\left[f(s_h,a_h)\right] - \mathbb{E}_{(s_h,a_h) \sim d_h^\mu}\left[f(s_h,a_h)\right]\right| \\
&\leq R\sqrt{\frac{2\log(8/\delta)}{N}}.
\end{aligned}$$

Consider union bound over all $h \in [H], f \in \mathcal{F}$, and $\pi \in \Pi$. Since $\pi_h^t \in \Pi$ for every $(t,h) \in [T] \times [H]$, with probability at least $1 - \delta/4$, we have

$$\begin{aligned}
&\left|\sum_{h=1}^H \mathbb{E}_{(s_h,a_h) \sim \mathcal{D}_{\text{traj}}}\left[f_h \circ \pi_h^t(s_h) - f_h(s_h,a_h)\right] - \sum_{h=1}^H \mathbb{E}_{(s_h,a_h) \sim d_h^\mu}\left[f_h \circ \pi_h^t(s_h) - f_h(s_h,a_h)\right]\right| \\
&\leq \sum_{h=1}^H \left|\mathbb{E}_{(s_h,a_h) \sim \mathcal{D}_{\text{traj}}}\left[f_h \circ \pi_h^t(s_h) - f_h(s_h,a_h)\right] - \mathbb{E}_{(s_h,a_h) \sim d_h^\mu}\left[f_h \circ \pi_h^t(s_h) - f_h(s_h,a_h)\right]\right| \\
&\leq RH\sqrt{\frac{2\log(8H|\mathcal{F}||\Pi|/\delta)}{N}} \leq 2RH\sqrt{\frac{T\log(8H|\mathcal{F}|/\delta)}{N}}.
\end{aligned}$$

for every $f \in \mathcal{F}$.

Step 2. We have that

$$|\hat{\mathcal{E}}_{\mathcal{D}_{\text{traj}}}(f;\hat{P},\hat{r}) - \mathcal{E}(f;P^\star,\hat{r})| \leq |\hat{\mathcal{E}}_{\mathcal{D}_{\text{traj}}}(f;\hat{P},\hat{r}) - \mathcal{E}(f;\hat{P},\hat{r})| + |\mathcal{E}(f;\hat{P},\hat{r}) - \mathcal{E}(f;P^\star,\hat{r})|. \quad (12)$$

Again, we use Hoeffding inequality to bound the first term. Fix $f \in \mathcal{F}^H$ and $\pi = \{\pi_h\}_{h=1}^H \in \Pi^H$ and consider the function $r^\pi_{\hat{P},f}$ (Recall that $r^\pi_{h,\hat{P},f}(s,a) = f_h(s,a) - \hat{P}(f_{h+1} \circ \pi_{h+1})(s,a)$ for all $h \in [H]$ and $(s,a) \in \mathcal{S} \times \mathcal{A}$). Since $|(r^\pi_{\hat{P},f} - \hat{r})(\tau)| \leq 2RH$ for any trajectory τ, we have that

$$\begin{aligned}
&\left|\mathbb{E}_{(\tau^0,\tau^1) \sim \mathcal{D}_{\text{traj}}}\left[\left|(r^\pi_{\hat{P},f} - \hat{r})(\tau^0) - (r^\pi_{\hat{P},f} - \hat{r})(\tau^1)\right|\right] - \mathbb{E}_{(\tau^0,\tau^1) \sim \mu}\left[\left|(r^\pi_{\hat{P},f} - \hat{r})(\tau^0) - (r^\pi_{\hat{P},f} - \hat{r})(\tau^1)\right|\right]\right| \\
&\leq 2RH\sqrt{\frac{2\log(8/\delta)}{N}}
\end{aligned}$$

with probability at least $1-\delta/8$. Applying union bound over all $f \in \mathcal{F}^H$ and $\pi \in \Pi^H$, since $\pi_h^t \in \Pi$ for every $(t,h) \in [T] \times [H]$, it holds that

$$|\hat{\mathcal{E}}_{\mathcal{D}_{\text{traj}}}(f;\hat{P},\hat{r}) - \mathcal{E}(f;\hat{P},\hat{r})| \leq 2RH\sqrt{\frac{2H\log(8|\mathcal{F}||\Pi|/\delta)}{N}} \leq 4RH\sqrt{\frac{HT\log(8|\mathcal{F}|/\delta)}{N}} \quad (13)$$

for every $f \in \mathcal{F}$, with probability at least $1 - \delta/8$.

On the other hand, the second term in (12) is bounded by

$$
\begin{aligned}
&|\mathcal{E}(f;\hat{P},\hat{r}) - \mathcal{E}(f;P^\star,\hat{r})| \\
&\le \mathbb{E}_{(\tau^0,\tau^1)\sim\mu}\left[\left|\sum_{h=1}^{H}(P^\star-\hat{P})(f_h\circ\pi_h^t)(s_h^0,a_h^0) - \sum_{h=1}^{H}(P^\star-\hat{P})(f_h\circ\pi_h^t)(s_h^1,a_h^1)\right|\right] \\
&\le \mathbb{E}_{\tau^0\sim\mu}\left[\sum_{h=1}^{H}\left|(P^\star-\hat{P})(f_h\circ\pi_h^t)(s_h^0,a_h^0)\right|\right] + \mathbb{E}_{\tau^1\sim\mu}\left[\sum_{h=1}^{H}\left|(P^\star-\hat{P})(f_h\circ\pi_h^t)(s_h^1,a_h^1)\right|\right] \\
&= 2\mathbb{E}_{\tau\sim\mu}\left[\sum_{h=1}^{H}\left|(P^\star-\hat{P})(f_h\circ\pi_h^t)(s_h,a_h)\right|\right] \\
&\le 2R\mathbb{E}_{\tau\sim\mu}\left[\sum_{h=1}^{H}\left\|P^\star(\cdot\mid s_h,a_h)-\hat{P}(\cdot\mid s_h,a_h)\right\|_1\right] \\
&= 2R\sum_{h=1}^{H}\mathbb{E}_{(s_h,a_h)\sim d_h^\mu}\left[\left\|P^\star(\cdot\mid s_h,a_h)-\hat{P}(\cdot\mid s_h,a_h)\right\|_1\right]
\end{aligned}
$$

where the first inequality holds since we have $||a|-|b|| \le |a-b|$ for all $a,b\in\mathbb{R}$, and the third inequality holds due to Hölder's inequality with the fact that $\|f_h\circ\pi_h^t\|_\infty \le R$. Furthermore, Lemma E.3 implies

$$
\begin{aligned}
|\mathcal{E}(f;\hat{P},\hat{r}) - \mathcal{E}(f;P^\star,\hat{r})| &\le 2R\sum_{h=1}^{H}\mathbb{E}_{(s_h,a_h)\sim d_h^\mu}\left[\left\|P^\star(\cdot\mid s_h,a_h)-\hat{P}(\cdot\mid s_h,a_h)\right\|_1\right] \\
&\le 2R\sum_{h=1}^{H}\sqrt{\mathbb{E}_{(s_h,a_h)\sim d_h^\mu}\left[\left\|P^\star(\cdot\mid s_h,a_h)-\hat{P}(\cdot\mid s_h,a_h)\right\|_1^2\right]} \\
&\le 2RH\epsilon_P(\delta/8) \qquad (14)
\end{aligned}
$$

with probability at least $1-\delta/8$. Taking union bound of the two event (13) and (14), with probability at least $1-\delta/4$, it holds that

$$
|\hat{\mathcal{E}}_{\mathcal{D}_{\text{traj}}}(f;\hat{P},\hat{r}) - \mathcal{E}(f;P^\star,\hat{r})| \le 2RH\sqrt{\frac{HT\log(8|\mathcal{F}|/\delta)}{N}} + 2RH\epsilon_P(\delta/8)
$$

for every $f\in\mathcal{F}$.

Finally, we conclude the proof by combining the bounds in Step 1 and Step 2. With probability at least $1-\delta/2$, for every $f\in\mathcal{F}$, it hols that

$$
\begin{aligned}
&\left|\hat{\mathcal{L}}_{\text{opt}}^t(f;\hat{P},\hat{r}) - \mathcal{L}_{\text{opt}}^t(f;\hat{r})\right| \\
&\le \left|\sum_{h=1}^{H}\mathbb{E}_{(s_h,a_h)\sim\mathcal{D}_{\text{traj}}}\left[f_h\circ\pi_h^t(s_h)-f_h(s_h,a_h)\right] - \sum_{h=1}^{H}\mathbb{E}_{(s_h,a_h)\sim d_h^\mu}\left[f_h\circ\pi_h^t(s_h)-f_h(s_h,a_h)\right]\right| \\
&\quad + \left|\hat{\mathcal{E}}_{\mathcal{D}_{\text{traj}}}(f;\hat{P},\hat{r}) - \mathcal{E}(f;P^\star,\hat{r})\right| \\
&\le 4RH\sqrt{\frac{T\log(8H|\mathcal{F}|/\delta)}{N}} + 2RH\sqrt{\frac{HT\log(8|\mathcal{F}|/\delta)}{N}} + 2RH\epsilon_P(\delta/8) \\
&\le 8R\sqrt{\frac{H^3T\log(8H|\mathcal{F}|/\delta)}{N}} + 2RH\epsilon_P(\delta/8).
\end{aligned}
$$

□

D.2 Policy Update

The analysis of the policy update step in Algorithm 2 follows the same argument in Lemma C.5.

Lemma D.3. *For any sequence of functions $\{f^t\}_{t=1}^T$, the policy update (Line 7) in Algorithm 2 guarantees that*

$$\frac{1}{T}\sum_{t=1}^{T}\left(V_{1,r^t}^{\pi^\star}(s_1) - V_{1,r^t}^{\pi^t}(s_1)\right) \le RH\sqrt{\frac{\log|\mathcal{A}|}{2T}}$$

where $r^t = r_{P^\star,f^t}^{\pi^t}$, i.e. $r_h^t(s,a) = f_h^t(s,a) - P^\star(f_{h+1}^t \circ \pi_{h+1}^t)(s,a)$ for all $h \in [H]$ and $(s,a) \in \mathcal{S}\times\mathcal{A}$.

Proof. Since we have the Bellman equation $f_h^t = r_h^t + P_h^\star(f_{h+1}^t \circ \pi_{h+1}^t)$ for all $h \in [H]$, we can apply the performance difference lemma (Lemma E.1) to obtain

$$\sum_{t=1}^{T}\left(V_{1,r^t}^{\pi^\star}(s_1) - V_{1,r^t}^{\pi^t}(s_1)\right) = \sum_{t=1}^{T}\sum_{h=1}^{H}\mathbb{E}_{\pi^\star}\left[\langle f_h^t(s_h,\cdot), \pi_h^\star(\cdot \mid s_h) - \pi_h^t(\cdot \mid s_h)\rangle\right].$$

Rearranging the inner product term, we see that

$$\begin{aligned}
&\langle \eta f_h^t(s_h,\cdot), \pi_h^\star(\cdot \mid s) - \pi_h^t(\cdot \mid s)\rangle \\
&\langle \eta f_h^t(s_h,\cdot), \pi_h^\star(\cdot \mid s) - \pi_h^{t+1}(\cdot \mid s)\rangle + \langle \eta f_h^t(s_h,\cdot), \pi_h^{t+1}(\cdot \mid s) - \pi_h^t(\cdot \mid s)\rangle \\
&\le \langle \eta f_h^t(s_h,\cdot), \pi_h^\star(\cdot \mid s) - \pi_h^{t+1}(\cdot \mid s)\rangle + \eta\left\|f_h^t(s_h,\cdot)\right\|_\infty\left\|\pi_h^\star(\cdot \mid s) - \pi_h^{t+1}(\cdot \mid s)\right\|_1 \\
&\le \langle \eta f_h^t(s_h,\cdot), \pi_h^\star(\cdot \mid s) - \pi_h^{t+1}(\cdot \mid s)\rangle + \eta R\left\|\pi_h^\star(\cdot \mid s) - \pi_h^{t+1}(\cdot \mid s)\right\|_1 \qquad (15)
\end{aligned}$$

where we use Hölder's inequality with the fact that $\|f_h^t\|_\infty \le R$. Now recall that the policy update step in Algorithm 3 leads to

$$\pi_h^{t+1}(\cdot \mid s) = \frac{1}{Z_h^t(s)}\pi_h^t(\cdot \mid s)\exp\left(\eta f_h^t(s,\cdot)\right)$$

where $Z_h^t(s) = \sum_{a\in\mathcal{A}}\pi_h^t(a \mid s)\exp\left(\eta f_h^t(s,a)\right)$. Using the relationship $\eta f_h^t(s,a) = \log Z_h^t(s) + \log\pi_h^{t+1}(a \mid s) - \log\pi_h^t(a \mid s)$, it holds that

$$\begin{aligned}
&\langle \eta f_h^t(s_h,\cdot), \pi_h^\star(\cdot \mid s) - \pi_h^{t+1}(\cdot \mid s)\rangle \\
&= \langle \log Z_h^t(s) + \log\pi_h^{t+1}(\cdot \mid s) - \log\pi_h^t(\cdot \mid s), \pi_h^\star(\cdot \mid s) - \pi_h^{t+1}(\cdot \mid s)\rangle \\
&= \langle \log\pi_h^{t+1}(\cdot \mid s) - \log\pi_h^t(\cdot \mid s), \pi^\star(\cdot \mid s) - \pi_h^{t+1}(\cdot \mid s)\rangle \\
&= \langle \log\pi_h^{t+1}(\cdot \mid s) - \log\pi_h^t(\cdot \mid s), \pi^\star(\cdot \mid s)\rangle - D_{KL}\left(\pi_h^{t+1}(\cdot \mid s)\|\pi_h^t(\cdot \mid s)\right) \\
&= \langle \log\frac{\pi_h^{t+1}(\cdot \mid s)}{\pi_h^\star(\cdot \mid s)} + \log\frac{\pi_h^\star(\cdot \mid s)}{\pi_h^t(\cdot \mid s)}, \pi_h^\star(\cdot \mid s)\rangle - D_{KL}\left(\pi_h^{t+1}(\cdot \mid s)\|\pi_h^t(\cdot \mid s)\right) \\
&= D_{KL}\left(\pi_h^\star(\cdot \mid s)\|\pi_h^t(\cdot \mid s)\right) - D_{KL}\left(\pi_h^\star(\cdot \mid s)\|\pi_h^{t+1}(\cdot \mid s)\right) - D_{KL}\left(\pi_h^{t+1}(\cdot \mid s)\|\pi_h^t(\cdot \mid s)\right) \\
&\le D_{KL}\left(\pi_h^\star(\cdot \mid s)\|\pi_h^t(\cdot \mid s)\right) - D_{KL}\left(\pi_h^\star(\cdot \mid s)\|\pi_h^{t+1}(\cdot \mid s)\right) - \frac{1}{2}\left\|\pi_h^\star(\cdot \mid s) - \pi_h^{t+1}(\cdot \mid s)\right\|_1^2
\end{aligned}$$

where the second equality holds since $Z_h^t(s)$ is a constant given s, and the last inequality holds due to Pinsker's inequality. Combining this bound with (15), we obtain

$$
\begin{aligned}
&\sum_{t=1}^{T} \langle \eta f_h^t(s_h, \cdot), \pi_h^\star(\cdot \mid s) - \pi_h^t(\cdot \mid s) \rangle \\
&= \sum_{t=1}^{T} \left(D_{KL}\left(\pi_h^\star(\cdot \mid s) \| \pi_h^t(\cdot \mid s)\right) - D_{KL}\left(\pi_h^\star(\cdot \mid s) \| \pi_h^{t+1}(\cdot \mid s)\right)\right) \\
&\quad + \sum_{t=1}^{T} \left(\eta R \left\| \pi_h^\star(\cdot \mid s) - \pi_h^{t+1}(\cdot \mid s) \right\|_1 - \frac{1}{2} \left\| \pi_h^\star(\cdot \mid s) - \pi_h^{t+1}(\cdot \mid s) \right\|_1^2 \right) \\
&\leq \sum_{t=1}^{T} \left(D_{KL}\left(\pi_h^\star(\cdot \mid s) \| \pi_h^t(\cdot \mid s)\right) - D_{KL}\left(\pi_h^\star(\cdot \mid s) \| \pi_h^{t+1}(\cdot \mid s)\right)\right) + \sum_{t=1}^{T} \frac{\eta^2 R^2}{2} \\
&= D_{KL}\left(\pi_h^\star(\cdot \mid s) \| \pi_h^1(\cdot \mid s)\right) - D_{KL}\left(\pi_h^\star(\cdot \mid s) \| \pi_h^{T+1}(\cdot \mid s)\right) + \frac{\eta^2 R^2 T}{2} \\
&\leq \log |\mathcal{A}| + \frac{\eta^2 R^2 T}{2}
\end{aligned}
$$

where the first inequality holds since $\forall x \in \mathbb{R}\, ax - x^2/2 \leq a^2/2$, and the second inequality holds due to the fact that $\pi_h^1 = \text{Unif}(\mathcal{A})$. Finally, setting $\eta = \sqrt{\frac{2 \log |\mathcal{A}|}{R^2 T}}$, we have

$$
\begin{aligned}
\sum_{t=1}^{T} V_{1,r^t}^{\pi^\star} - V_{1,r^t}^{\pi^t} &= \sum_{t=1}^{T} \mathbb{E}_{\pi^\star} \left[\sum_{h=1}^{H} \langle f_h^t(s_h, \cdot), \pi_h^\star(\cdot \mid s_h) - \pi_h^t(\cdot \mid s_h) \rangle \right] \\
&= \sum_{h=1}^{H} \mathbb{E}_{\pi^\star} \left[\sum_{t=1}^{T} \langle f_h^t(s_h, \cdot), \pi_h^\star(\cdot \mid s_h) - \pi_h^t(\cdot \mid s_h) \rangle \right] \\
&\leq \sum_{h=1}^{H} \left(\frac{\log |\mathcal{A}|}{\eta} + \frac{\eta R^2 T}{2} \right) = RH \sqrt{\frac{T \log |\mathcal{A}|}{2}}.
\end{aligned}
$$

□

Finally, we prove Theorem 4.1.

Proof of Theorem 4.1. For simplicity, we write $r^t = r_{P^\star, f^t}^{\pi^t}$, i.e. $r_h^t(s,a) = f_h^t(s,a) - P_h^\star(f_{h+1}^t \circ \pi_{h+1}^t)(s,a)$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$ and $h \in [H]$. The condition $r^t \in \mathcal{R}^H$ is not required; we only rely on the boundedness $\|r_h^t\|_\infty \leq R$ for all h, which Assumption 3 guarantees.

Condition on the events in Lemma E.2 (with $\delta' = \delta/2$) and Lemma D.1, which hold simultaneously with probability at least $1 - \delta$. Consider the following sub-optimality decomposition at step t:

$$
\begin{aligned}
V_{1,r^\star}^{\pi^\star} - V_{1,r^\star}^{\pi^t} &= V_{1,r^\star}^{\pi^\star} - V_{1,\hat{r}}^{\pi^\star} + V_{1,\hat{r}}^{\pi^\star} - V_{1,r^t}^{\pi^\star} + V_{1,r^t}^{\pi^\star} - V_{1,r^\star}^{\pi^t} + V_{1,r^t}^{\pi^t} - V_{1,r^t}^{\pi^t} \\
&= \underbrace{V_{1,r^\star - \hat{r}}^{\pi^\star} - V_{1,r^\star - \hat{r}}^{\mu}}_{\text{(I) : MLE estimation error}} \\
&\quad + \underbrace{V_{1,\hat{r} - r^t}^{\pi^\star} - V_{1,\hat{r} - r^t}^{\mu} - V_{1,r^\star}^{\pi^t} + V_{1,r^\star}^{\mu} + V_{1,r^t}^{\pi^t} - V_{1,r^t}^{\mu}}_{\text{(II) : Optimization error}} \\
&\quad + \underbrace{V_{1,r^t}^{\pi^\star} - V_{1,r^t}^{\pi^t}}_{\text{(III) : Policy update regret}} ,
\end{aligned} \tag{16}
$$

where we omit the initial state s_1 for simplicity.

Bounding (I). Using Lemma E.2, the MLE estimation error is bounded by:

$$\begin{aligned}
\text{(I)} &= V^{\pi^\star}_{1,r^\star-\hat{r}} - V^{\mu}_{1,r^\star-\hat{r}} \\
&= \mathbb{E}_{\tau^0\sim\pi^\star,\tau^1\sim\mu}\left[r^\star(\tau^0) - r^\star(\tau^1) - \hat{r}(\tau^0) + \hat{r}(\tau^1)\right] \\
&\leq \sqrt{\mathbb{E}_{\tau^0\sim\pi^\star,\tau^1\sim\mu}\left[|r^\star(\tau^0) - r^\star(\tau^1) - \hat{r}(\tau^0) + \hat{r}(\tau^1)|^2\right]} \\
&\leq \sqrt{C_{\text{TR}}\mathbb{E}_{\tau^0,\tau^1\sim\mu}\left[|r^\star(\tau^0) - r^\star(\tau^1) - \hat{r}(\tau^0) + \hat{r}(\tau^1)|^2\right]} \\
&\leq \sqrt{C_{\text{TR}}}\epsilon_r(\delta/2)
\end{aligned}$$

Bounding (II). We can relate the terms $V^{\pi^\star}_{1,\hat{r}-r^t} - V^{\mu}_{1,\hat{r}-r^t}$ to the trajectory-pair ℓ_1 loss $\mathcal{E}(f^t; P^\star, \hat{r})$. By Assumption 4, we have that

$$\begin{aligned}
&V^{\pi^\star}_{1,\hat{r}-r^t} - V^{\mu}_{1,\hat{r}-r^t} \\
&= \mathbb{E}_{\tau^0\sim\pi^\star,\tau^1\sim\mu}\left[\hat{r}(\tau^0) - \hat{r}(\tau^1) - r^t(\tau^0) + r^t(\tau^1)\right] \\
&\leq C_{\text{TR}}\mathbb{E}_{\tau^0,\tau^1\sim\mu}\left[|\hat{r}(\tau^0) - \hat{r}(\tau^1) - r^t(\tau^0) + r^t(\tau^1)|\right] \\
&= C_{\text{TR}}\mathbb{E}_{\tau^0,\tau^1\sim\mu}\left[|r^{\pi^t}_{P^\star,f^t}(\tau^0) - r^{\pi^t}_{P^\star,f^t}(\tau^1) - \hat{r}(\tau^0) + \hat{r}(\tau^1)|\right] \\
&= C_{\text{TR}}\mathcal{E}(f^t; P^\star, \hat{r}) \leq \lambda\mathcal{E}(f^t; P^\star, \hat{r})
\end{aligned}$$

where the last inequality holds since $\mathcal{E}(f^t; P^\star, \hat{r})$ is non-negative and $\lambda \geq C_{\text{TR}}$. Further, Lemma D.1 and Lemma E.1 implies

$$\begin{aligned}
\lambda\mathcal{E}(f^t; P^\star, \hat{r}) &\leq \sum_{h=1}^{H}\mathbb{E}_{(s_h,a_h)\sim d^\mu_h}\left[Q^{\pi^t}_h \circ \pi^t_h(s_h) - Q^{\pi^t}_h(s_h, a_h)\right] + \lambda\mathcal{E}(Q^{\pi^t}; P^\star, \hat{r}) \\
&\quad - \sum_{h=1}^{H}\mathbb{E}_{(s_h,a_h)\sim d^\mu_h}\left[f^t_h \circ \pi^t_h(s_h) - f^t_h(s_h, a_h)\right] + 2\epsilon_{approx} \\
&= \left(V^{\pi^t}_{1,r^\star} - V^{\mu}_{1,r^\star}\right) + \lambda\mathcal{E}(Q^{\pi^t}; P^\star, \hat{r}) - \left(V^{\pi^t}_{1,r^t} - V^{\mu}_{1,r^t}\right) + 2\epsilon_{approx}.
\end{aligned}$$

On the other hand, note that

$$\begin{aligned}
r^{\pi^t}_{P^\star,Q^{\pi^t}}(\tau) &= \sum_{h=1}^{H}\left(Q^{\pi^t}_h(s_h, a_h) - P^\star(Q^{\pi^t}_{h+1} \circ \pi^t_{h+1})(s_h, a_h)\right) \\
&= \sum_{h=1}^{H}\left(Q^{\pi^t}_h(s_h, a_h) - P^\star V^{\pi^t}_{h+1}(s_h, a_h)\right) \\
&= \sum_{h=1}^{H} r^\star_h(s_h, a_h) = r^\star(\tau)
\end{aligned}$$

for any $\tau = (s_1, a_1, \ldots, s_H, a_H)$, i.e. $r^{\pi^t}_{P^\star,Q^{\pi^t}} = r^\star$. Thus, we have

$$\begin{aligned}
\lambda\mathcal{E}(Q^{\pi^t}; P^\star, \hat{r}) &= \lambda\mathbb{E}_{(\tau_0,\tau_1)\sim\mu}\left[\left|\{r^\star(\tau^0) - r^\star(\tau^1)\} - \{\hat{r}(\tau^0) - \hat{r}(\tau^1)\}\right|\right] \\
&\leq \lambda\sqrt{\mathbb{E}_{(\tau_0,\tau_1)\sim\mu}\left[\left|\{r^\star(\tau^0) - r^\star(\tau^1)\} - \{\hat{r}(\tau^0) - \hat{r}(\tau^1)\}\right|^2\right]} \leq \lambda\epsilon_r(\delta/2)
\end{aligned}$$

where the inequality follows from Lemma E.2. Combining the results, we obtain

$$\begin{aligned}
\text{(II)} &= \left(V^{\pi^\star}_{1,\hat{r}-r^t} - V^{\mu}_{1,\hat{r}-r^t}\right) - \left(V^{\pi^t}_{1,r^\star} - V^{\mu}_{1,r^\star}\right) + \left(V^{\pi^t}_{1,r^t} - V^{\mu}_{1,r^t}\right) \\
&\leq \lambda\mathcal{E}(Q^{\pi^t}; P^\star, \hat{r}) + 2\epsilon_{approx} \leq \lambda\epsilon_r(\delta/2) + 2\epsilon_{approx}
\end{aligned}$$

Bounding Sub-optimality. Finally, we bound the sub-optimality $V_{1,r^\star}^{\pi^\star} - V_{1,r^\star}^{\hat{\pi}}$. Putting the bounds on (I) and (II) into (16), we have

$$\begin{aligned}
&V_{1,r^\star}^{\pi^\star} - V_{1,r^\star}^{\pi^t} \\
&\leq \sqrt{C_{\text{TR}}}\epsilon_r(\delta/2) + \lambda\epsilon_r(\delta/2) + 2\epsilon_{approx} + V_{1,r^t}^{\pi^\star} - V_{1,r^t}^{\pi^t}
\end{aligned}$$

Since Algorithm 2 returns the mixture policy $\hat{\pi} = \frac{1}{T}\sum_{t=1}^{T}\pi^t$, the sub-optimality is $V_{1,r^\star}^{\pi^\star} - V_{1,r^\star}^{\hat{\pi}} = \frac{1}{T}\sum_{t=1}^{T}\left(V_{1,r^\star}^{\pi^\star} - V_{1,r^\star}^{\pi^t}\right)$. Using the bounds we derived and Lemma D.3, it holds that

$$\begin{aligned}
&V_{1,r^\star}^{\pi^\star} - V_{1,r^\star}^{\hat{\pi}} \\
&= \frac{1}{T}\sum_{t=1}^{T}\left(V_{1,r^\star}^{\pi^\star} - V_{1,r^\star}^{\pi^t}\right) \\
&\leq \sqrt{C_{\text{TR}}}\epsilon_r(\delta/2) + \lambda\epsilon_r(\delta/2) + 2\epsilon_{approx} + \frac{1}{T}\sum_{t=1}^{T}\left(V_{1,r^t}^{\pi^\star} - V_{1,r^t}^{\pi^t}\right) \\
&\leq \sqrt{C_{\text{TR}}}\epsilon_r(\delta/2) + \lambda\epsilon_r(\delta/2) + 2\epsilon_{approx} + RH\sqrt{\frac{\log|\mathcal{A}|}{2T}} \\
&\leq \mathcal{O}\left(C_{\text{TR}}\sqrt{\frac{\kappa^2 H\log(|\mathcal{R}|/\delta)}{M}} + R\sqrt{\frac{H^3 T\log(H|\mathcal{F}|/\delta)}{N}} + RH\sqrt{\frac{\log(H|\mathcal{P}|/\delta)}{N}} + RH\sqrt{\frac{\log|\mathcal{A}|}{T}}\right) \\
&\leq \mathcal{O}\left(C_{\text{TR}}\sqrt{\frac{\kappa^2 H\log(|\mathcal{R}|/\delta)}{M}} + RH\sqrt{\frac{\max\{HT\log(H|\mathcal{F}|/\delta), \log(H|\mathcal{P}|/\delta)\}}{N}} + RH\sqrt{\frac{\log|\mathcal{A}|}{T}}\right).
\end{aligned}$$

□

E SUPPORTING LEMMAS

Lemma E.1 (Performance Difference Lemma (Kakade & Langford, 2002))**.** *Let P be any transition probability, and denote the corresponding value function by V. Let $\pi, \tilde{pi}$ be any policies. For any reward r, we have that*

$$V_{1,r}^{\pi}(s_1) - V_{1,r}^{\tilde{\pi}}(s_1) = \sum_{h=1}^{H}\mathbb{E}_{s_h\sim d_h^\pi}\left[\langle Q_{h,r}^{\tilde{\pi}}(s_h,\cdot), \pi(\cdot\mid s_h) - \tilde{\pi}(\cdot\mid s_h)\rangle\right]$$

Proof. Recursively applying the Bellman equation, we obtain

$$\begin{aligned}
V_{1,r}^{\pi}(s_1) - V_{1,r}^{\tilde{\pi}}(s_1) &= \mathbb{E}_\pi[r(s_1,a_1) + V_{2,r}^{\pi}(s_2)] - \mathbb{E}_\pi[V_{1,r}^{\tilde{\pi}}(s_1)] \\
&= \mathbb{E}_\pi[Q_{1,r}^{\tilde{\pi}}(s_1,a_1) - V_{2,r}^{\tilde{\pi}}(s_2) + V_{2,r}^{\pi}(s_2)] - \mathbb{E}_\pi[V_{1,r}^{\tilde{\pi}}(s_1)] \\
&= \mathbb{E}_\pi[Q_{1,r}^{\tilde{\pi}}(s_1,a_1) - V_{1,r}^{\tilde{\pi}}(s_1)] + \mathbb{E}_\pi[V_{2,r}^{\pi}(s_2) - V_{2,r}^{\tilde{\pi}}(s_2)] \\
&= \mathbb{E}_\pi[\langle Q_{1,r}^{\tilde{\pi}}(s_1,\cdot), \pi(\cdot\mid s_1) - \tilde{\pi}(\cdot\mid s_1)\rangle] + \mathbb{E}_\pi[V_{2,r}^{\pi}(s_2) - V_{2,r}^{\tilde{\pi}}(s_2)] \\
&= \cdots \\
&= \sum_{h=1}^{H}\mathbb{E}_{s_h\sim d_h^\pi}\left[\langle Q_{h,r}^{\tilde{\pi}}(s_h,\cdot), \pi(\cdot\mid s_h) - \tilde{\pi}(\cdot\mid s_h)\rangle\right].
\end{aligned}$$

□

Lemma E.2 (Lemma 2 in Zhan et al. (2024a))**.** *With probability at least $1-\delta'$, we have*

$$\mathbb{E}_{\tau^0,\tau^1\sim\mu}\left[|(\hat{r}(\tau^0) - \hat{r}(\tau^1)) - (r^\star(\tau^0) - r^\star(\tau^1))|^2\right] \leq \frac{c_1\kappa^2 H\log(|\mathcal{R}|/\delta')}{M} := \epsilon_r^2(\delta')$$

Lemma E.3 (Lemma 3 in Zhan et al. (2024a)). *With probability at least $1-\delta'$, for all $h \in [H]$, it holds that*

$$\mathbb{E}_{(s_h,a_h)\sim d_h^\mu}\left[\left\|\hat{P}_h(\cdot \mid s_h,a_h) - P^\star(\cdot \mid s_h,a_h)\right\|_1^2\right] \leq \frac{c_2 \log(H|\mathcal{P}|/\delta')}{N} := \epsilon_P^2(\delta')$$

where c_2 is an absolute constant.

Lemma E.4 (Lemma 15 in Song et al. (2023)). *Fix any $B > 0$, $\delta \in (0,1)$ and assume we have a class of real-valued functions $\mathcal{H} : \mathcal{X} \to [-B, B]$. Suppose we have K i.i.d. samples $\{(x_k, y_k)\}_{k=1}^K$ where $x_k \sim \rho$ and $y_k = h^\star(x_k) + \epsilon_k$ where $h^\star \in \mathcal{H}$ and $\{\epsilon_k\}_{k=1}^K$ are independent random variables such that $\mathbb{E}[\epsilon_k \mid x_k] = 0$. Additionally, suppose that $\max_k |y_k| \leq R$ and $\sup_{x\in\mathcal{X}} |h^\star(x)| \leq B$. Then, with probability at least $1-\delta$, the least square estimator $\hat{h} \in \arg\min_{h\in\mathcal{H}} \sum_{k=1}^K (h(x_k)-y_k)^2$ satisfies:*

$$\mathbb{E}_{x\sim\rho}\left[\left(\hat{h}(x) - h^\star(x)\right)^2\right] \leq \frac{c_2 B^2 \log(|\mathcal{H}|/\delta)}{K}$$

where c_2 is an absolute constant.

Algorithm 4 `APPO` (Practical version)

1: **Input:** Batch size B, Learning rates $\alpha_\phi, \alpha_\psi, \alpha_\theta$, constants $\lambda > 0$, $\tau \in (0,1)$
2: Train reward model $\hat{r}$ based on $\mathcal{D}_{\text{pref}}$ ▷ Use any reward learning method
3: **for** `step`$= 1, 2, \ldots$ **do**
4: Sample mini-batch of transition tuples $\mathcal{B}_{\text{tup}}$ and trajectory pairs $\mathcal{B}_{\text{traj}}$ from $\mathcal{D}_{\text{traj}}$
5: Train Q functions $\phi^i \leftarrow \phi^i - \alpha_\phi \nabla_{\phi^i} \mathcal{L}^\lambda_{\phi^i}(\mathcal{B}_{\text{tup}}, \mathcal{B}_{\text{traj}})$ for $i \in \{1,2\}$ (6)
6: Update target Q function $\bar{\phi}^i = (1-\tau)\bar{\phi}^i + \tau\phi^i$ for $i \in \{1,2\}$
7: Train V function $\psi \leftarrow \psi - \alpha_\psi \nabla_\psi \mathcal{L}_\psi(\mathcal{B}_{\text{tup}})$ (7)
8: Train actor $\theta \leftarrow \theta + \alpha_\theta \nabla_\theta \mathcal{L}_\theta(\mathcal{B}_{\text{tup}})$ (8)
9: **end for**

F ADDITIONAL EXPERIMENTS

F.1 EVALUATION ON THE META-WORLD `MEDIUM-EXPERT` DATASET

To further assess the generalization capability of `APPO`, we collected the Meta-World `medium-expert` dataset following the data collection procedures outlined in prior works (Hejna & Sadigh, 2024; Choi et al., 2024). Detailed information on the dataset is provided in Section G. For comparison, we use MR, the most effective baseline method identified in Table 1. The results in Table 2 show that `APPO` consistently outperforms or matches MR.

# of feedback	500		1000	
Dataset	dial-turn	sweep-into	dial-turn	sweep-into
MR	$15.80_{\pm 12.73}$	$14.32_{\pm 3.39}$	$26.08_{\pm 18.78}$	$8.48_{\pm 1.92}$
`APPO`	$32.40_{\pm 13.56}$	$12.80_{\pm 5.35}$	$39.20_{\pm 15.69}$	$14.56_{\pm 6.25}$

Table 2: Success rates on Meta-World `medium-expert` dataset with 500, 1000 preference feedback samples, averaged over 5 random seeds.

F.2 EFFECT OF DATASET SIZE

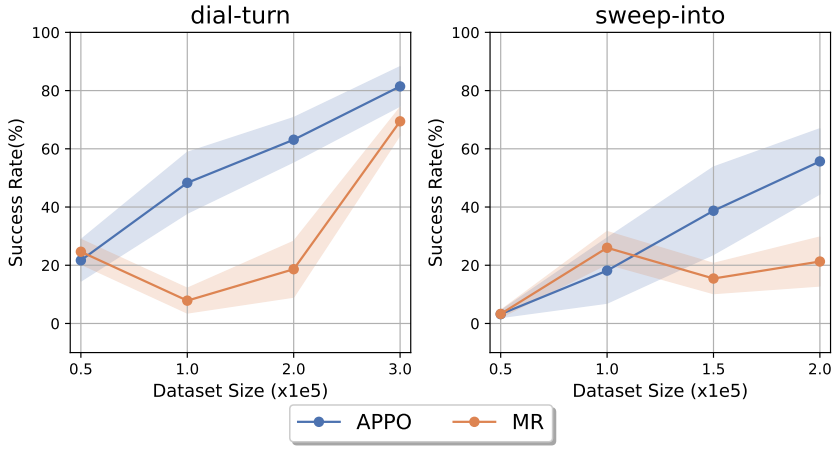

Figure 3: Success rates of `APPO` and MR evaluated in Meta-World `medium-replay` datasets, with varying dataset sizes. The number of preference feedback samples is fixed at 1000.

To examine the impact of dataset size $|\mathcal{D}_{\text{traj}}|$, we conducted experiments with varying sizes of the Meta-World `medium-replay` datasets. As shown in Figure 3, the performance of MR fluctuates with changes in dataset size, whereas the performance of `APPO` exhibits a more consistent and gradual response to dataset size variations.

F.3 LEARNING CURVES FROM EXPERIMENTS.

Figure 4 and Figure 5 show the learning curves of the experiments in Table 1 and Table 2. Each algorithm is trained for 250,000 gradient steps, with evaluations conducted every 5,000 steps. The success rates from the final five evaluation points are averaged and reported in Table 1 and Table 2.

Dataset	BPT	box-close	dial-turn	sweep	BPT-wall	sweep-into	drawer-open	lever-pull
Size ($\times 10^5$)	1.0	8.0	3.0	7.0	1.5	1.0	1.0	3.0

Table 3: The sizes of Meta-World `medium-replay` datasets (Choi et al., 2024). The abbreviation BPT indicates button-press-topdown.

G Experimental Details

G.1 Datasets

The Meta-World `medium-replay` dataset from Choi et al. (2024) consists of replay buffers generated by SAC (Haarnoja et al., 2018) agents with an approximate success rate of 50%. The dataset sizes are detailed in Table 3.

The Meta-World `medium-expert` dataset was collected following the procedures described in prior works (Hejna & Sadigh, 2024; Choi et al., 2024). Each dataset contains trajectories from five sources: (1) an expert policy, (2) expert policies for randomized variants and goals of the task, (3) expert policies for different tasks, (4) a random policy, and (5) an ϵ-greedy expert policy that takes greedy actions with a 50% probability. These trajectories are included in the dataset in proportions of $1:1:2:4:4$, respectively. Additionally, standard Gaussian noise was added to the actions of each policy. The dataset sizes match those of the `medium-replay` dataset.

G.2 Implementation and Hyperparameters.

For a fair comparison with baseline methods, we train the reward model and MR following the official implementation of Choi et al. (2024). The reward model is implemented by an ensemble model of three fully connected neural networks with three hidden layers, each containing 128 neurons. For critics (Q and V) and policies, we use fully connected neural networks with three hidden layers of 256 neurons each. Other hyperparameters are listed in Table 4. We find that using a lower learning rate for π and softer target network updates improves the stability of `APPO` training. Experiments were conducted on an Intel Xeon Gold 6226R CPU and an Nvidia GeForce RTX 3090 GPU. Each training session consists of 250,000 gradient steps, taking approximately 3-4 hours to complete. Our code is available at `https://github.com/oh-lab/APPO.git`.

Algorithm	Component	Value
Reward model	Neural networks	3-layers, hidden dimension 128
	Activation	ReLU for hidden activations, Tanh for final activation
	Optimizer	Adam (Kingma & Ba, 2015) with learning rate 1e-3
	Batch size	512
	Epochs	300
	Ensembles	3
MR	Neural networks (Q, V, π)	3-layers, hidden dimension 256
	Activaton	ReLU for hidden activations
	Q, V, π optimizer	Adam with learning rate 3e-4
	Batch size	256
	Target network soft update	0.005
	β (IQL advantage weight)	3.0
	τ (IQL expectile parameter)	0.7
	discount factor	0.99
APPO	Neural networks (Q, V, π)	3-layers, hidden dimension 256
	Activaton	LeakyReLU for hidden activations
	Q,V, α optimizer	Adam with learning rate 3e-4
	π optimizer	Adam with learning rate 3e-5
	Batch size	256 transitions and 16 trajectory pairs
	Target network soft update	0.001
	discount factor	0.99

Table 4: Implementation details and hyperparameters. For the reward model and MR algorithm, we follow the official implementation of Choi et al. (2024).

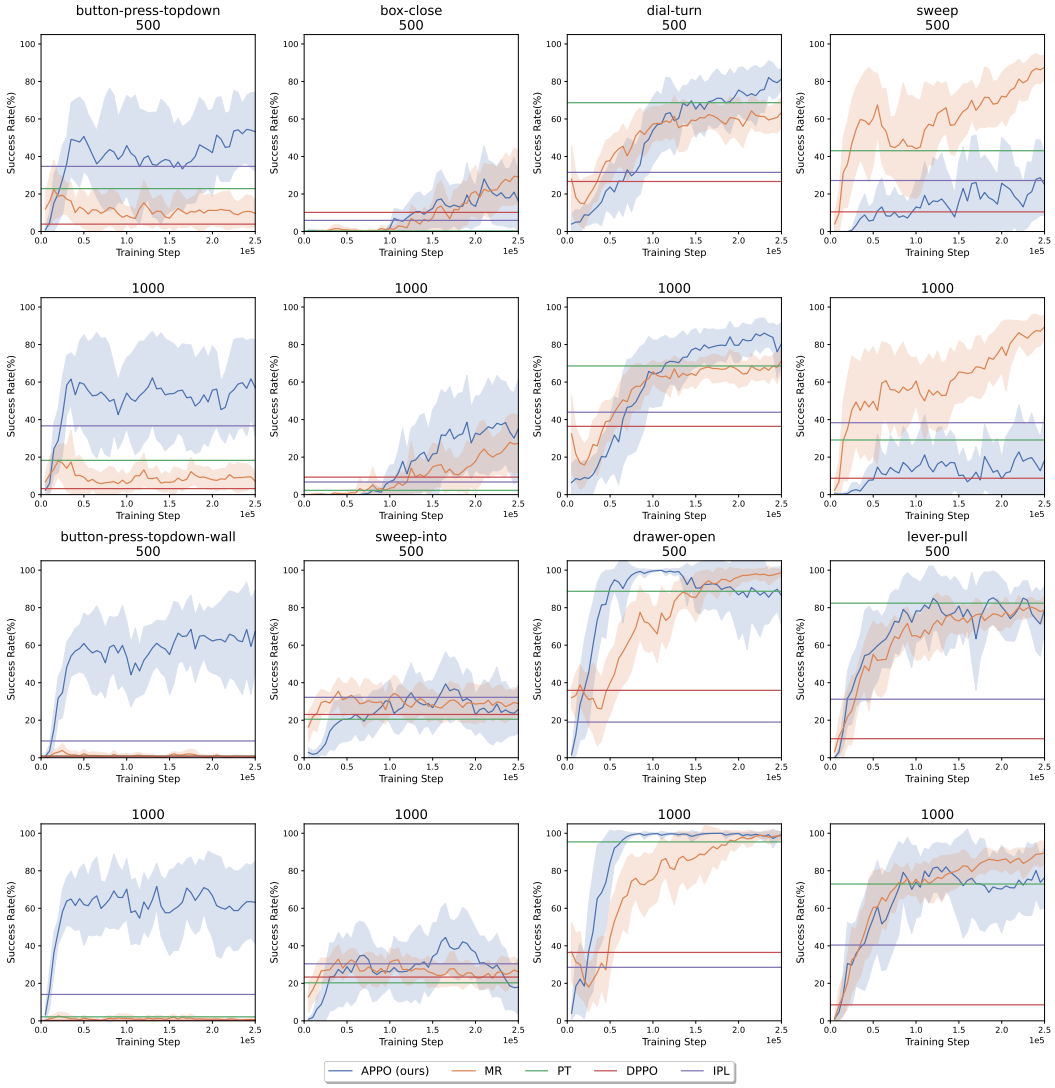

Figure 4: Learning Curves from the experiments in Table 1.

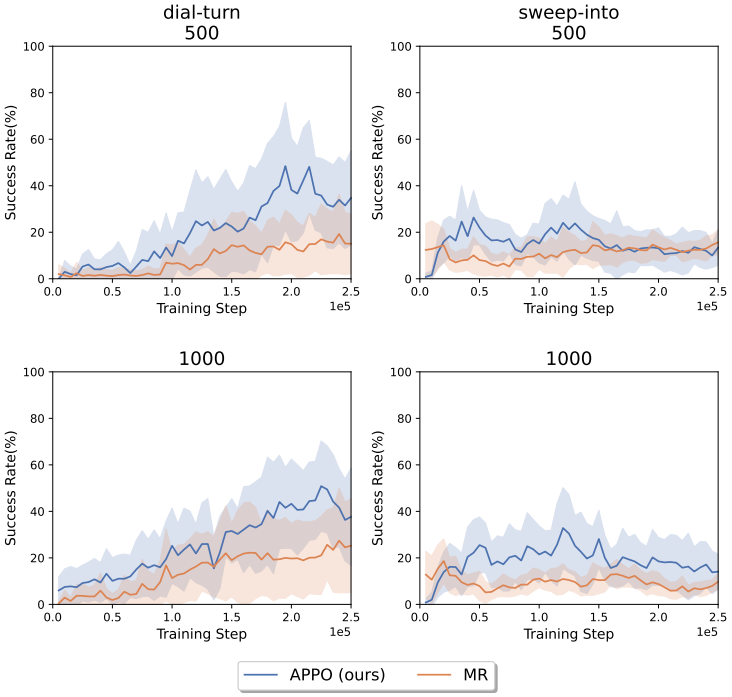

Figure 5: Learning Curves from the experiments in Table 2.