FEWER QUESTIONS, BETTER ANSWERS: EFFICIENT OFFLINE PREFERENCE-BASED REINFORCEMENT LEARNING VIA IN-DATASET EXPLORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Preference-based reinforcement learning (PbRL) can help avoid sophisticated reward designs and align better with human intentions, showing great promise in various real-world applications. However, obtaining human feedback for preferences can be expensive and time-consuming, which forms a strong barrier for PbRL. In this work, we address the problem of low query efficiency in offline PbRL, pinpointing two primary reasons: inefficient exploration and overoptimization of learned reward functions. In response to these challenges, we propose a novel algorithm, **O**ffline **P**b**R**L via **I**n-**D**ataset **E**xploration (OPRIDE), designed to enhance the query efficiency of offline PbRL. OPRIDE consists of two key features: a principled exploration strategy that maximizes the informativeness of the queries and a discount scheduling mechanism aimed at mitigating overoptimization of the learned reward functions. Through empirical evaluations, we demonstrate that OPRIDE significantly outperforms prior methods, achieving strong performance with notably fewer queries. Moreover, we provide theoretical guarantees of the algorithm's efficiency. Experimental results across various locomotion, manipulation, and navigation tasks underscore the efficacy and versatility of our approach.

1 INTRODUCTION

Reinforcement learning (RL) has proven effective across a range of sequential decision-making tasks, from mastering games like Go (Silver et al., 2016) to controlling complex systems such as robots (Ahn et al., 2022) and plasma reactors (Degrave et al., 2022). However, in many real-world applications, designing an appropriate reward function is a daunting challenge, as these tasks often involve objectives that are difficult to formalize with numerical rewards.

Preference-based RL (PbRL) (Akrour et al., 2012; Christiano et al., 2017) has emerged as a promising paradigm, leveraging human feedback in the form of pairwise preferences, which are inherently more interpretable yet still information-rich. This paradigm allows agents to learn from relative judgments rather than numerical reward signals, significantly reducing the complexity of reward design. Recent advancements in PbRL have illustrated its efficacy in enabling agents to learn novel behaviors (Christiano et al., 2017; Kim et al., 2023) and in achieving better alignment with human preferences (Ouyang et al., 2022), which are often difficult to encapsulate in a reward function. Despite these advantages, PbRL methods still face critical challenges, particularly in acquiring human feedback efficiently. Querying human preferences is both time-consuming and resource-intensive, limiting the scalability of PbRL in real-world applications.

To address this challenge, we propose **Offline PbRL via In-Dataset Exploration (OPRIDE)**, a novel algorithm designed to systematically enhance the query efficiency of offline PbRL, as depicted in Figure 1. OPRIDE introduces a principled exploration strategy that identifies the most informative queries by analyzing value differences between trajectories, ensuring that each query maximally contributes to learning the optimal policy. Additionally, to prevent overoptimization of the learned reward function (Gao et al., 2023; Zhu et al., 2024), particularly in regions with high uncertainty, we incorporate a discount factor scheduling mechanism that dynamically adjusts the discount based on the variance in the reward estimation. Based on the pessimistic property of the smaller discount

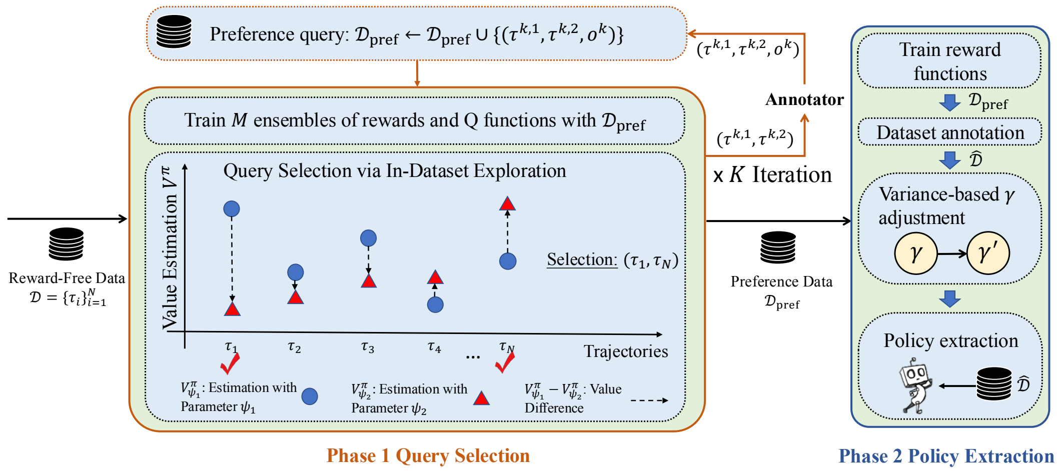

Figure 1: Framework of OPRIDE. The procedure consists of two phases. In the first offline phase, we select query based on exploration mechanism. The blue circles and red triangles represent the value estimation V_{ψ_1} and V_{ψ_2}, respectively. We select two trajectory pairs such that the disagreement of the preference between them is maximized (i.e., V_{ψ_1} strongly prefers τ_1 while V_{ψ_2} strongly prefers τ_N). In the second stage, we first learn an reward function based on the preference dataset and then annotate the reward-free dataset. Next, we adjust the discount factor based on the variance of the value function estimate to reduce the impact of noise in the reward learning.

factor, we can address the overestimation issue of the value function and, subsequently, a better policy performance and higher query efficiency.

Experimental evaluations on diverse locomotion and manipulation tasks, including AntMaze (Fu et al., 2020) and Meta-World (Yu et al., 2019), demonstrate the efficacy of our approach in achieving strong performance with significantly fewer queries compared to state-of-the-art baselines. Remarkably, our method achieves compelling results with as few as ten queries on Meta-World tasks, underscoring its efficiency and scalability. Furthermore, we provide theoretical insights into the efficiency of our algorithm, demonstrating that our exploration strategy is provably efficient under mild assumptions.

Our contributions are threefold: (1) We introduce OPRIDE, a novel offline PbRL algorithm that achieves superior query efficiency through in-dataset exploration; (2) We conduct extensive ablation studies that highlight the effectiveness of each component, providing insights into the factors driving query efficiency; and (3) We provide theoretical analyses establishing the provable efficiency of our algorithm involving a principled exploration strategy under mild assumptions.

1.1 Related Work

Preference-based RL. Various methods have been proposed to leverage human preferences (Akrour et al., 2012; Ibarz et al., 2018) and have demonstrated success in tackling complex control tasks (Christiano et al., 2017; Lee et al., 2021) and in aligning large language models (Stiennon et al., 2020; Ouyang et al., 2022; Rafailov et al., 2023; 2024). In the realm of offline Preference-based Reinforcement Learning (PbRL), a benchmark including several baselines (e.g., disagreement based method) is introduced by OPRL (Shin et al., 2023), which selects queries based on disagreement between the reward models and is inefficient in determining the optimal policy. Kim et al. (2023) apply Transformer models to effectively capture preferences for better credit assignment. Kang et al. (2023) present a direct approach to learning policy based on preferences. A recent work by Lindner et al. (2021) proposes an information-directed query selection method for PbRL, using the Laplacian approximation and the Hessian matrix for posterior computation. In contrast, our method selects queries to maximize the information gain about the optimal policy rather than the reward function, ensuring higher query efficiency.

In addition to empirical achievements, prior studies have also explored the theoretical aspects of PbRL. Pacchiano et al. (2021) propose a provable PbRL algorithm tailored for linear MDPs. Chen et al. (2022) extend this approach to scenarios where the Eluder dimension is finite. Zhan et al. (2023a) delve into the study of PbRL within an offline setting where a preference dataset is provided. Wang et al. (2023) propose an efficient randomized algorithm for PbRL in linear MDP and an efficient TS-based algorithm for nonlinear cases with finite Eluder dimensions. Sekhari et al. (2023) provides a PbRL algorithm with PAC guarantees. Novoseller et al. (2020) proposes the dueling posterior sampling algorithm that has an information-theoretic guarantee. Xu et al. (2020) provide a gap-dependent analysis for preference-based contextual bandit and imitation learning. Wu & Sun (2023) analyze the complexity of learning with utility-based preferences and general preferences.

Semi-supervised offline RL. In reward-free offline RL setting, Yu et al. (2022) and Hu et al. (2023) utilize reward-free data to aid offline learning, assuming the availability of a labeled offline dataset for reward function learning. Ajay et al. (2020) and Yang et al. (2023) utilize reward-free datasets by extracting valuable behaviors. Ye et al. and Ghosh et al. (2023) use reward-free offline data for pre-training, followed by online RL, where rewards are attainable. Ghosh et al. (2023) focuses on using reward-free offline data for representation learning, while Ye et al. explores the use of reward-free offline data for learning a latent dynamics model. In contrast, the offline PbRL setting will provide human feedback in the form of pairwise preference, which allows agents to learn from relative judgments rather than numerical reward signal.

RLHF. Reinforcement learning from human feedback (RLHF) has made significant strides following the outstanding success of ChatGPT (Team, 2024). This approach has emerged as a key method for aligning AI behavior more closely with human preferences, where PbRL (Christiano et al., 2017; Lee et al., 2021) plays a central role. This approach has been further refined and standardized in influential frameworks such as InstructGPT (Ouyang et al., 2022), Claude (Anthropic, 2023), and LLaMA2 (Touvron et al., 2023), etc.

2 PRELIMINARIES

We consider infinite-horizon Markov Decision Processes (MDPs), defined by the tuple $(\mathcal{S}, \mathcal{A}, \gamma, \mathcal{P}, r)$, with state space $\mathcal{S}$, action space $\mathcal{A}$, horizon H, transition function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ and reward function $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$. Without loss of generality, we assume a fixed start state s_0.

A policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ specifies a decision-making strategy in which the agent chooses actions adaptively based on the current state, that is, $a \sim \pi(\cdot \,|\, s)$. The value function $V^\pi : \mathcal{S} \to \mathbb{R}$ and the action-value function (Q-function) $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ are defined as

$$V^\pi(s) = \mathbb{E}_\pi\Big[\sum_{t=1}^{\infty} r(s_t, a_t) \,\Big|\, s_0 = s\Big], \quad Q^\pi(s, a) = \mathbb{E}_\pi\Big[\sum_{t=1}^{\infty} r(s_t, a_t) \,\Big|\, s_0 = s, a_0 = a\Big], \tag{1}$$

where the expectation is w.r.t. the trajectory τ induced by π.

We define the Bellman evaluation operator as

$$(\mathbb{T}^\pi f)(s, a) = \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a), a' \sim \pi(\cdot|s')}\big[r(s, a) + \gamma f(s', a')\big]. \tag{2}$$

We use $\pi^\star$, $Q^\star$, and $V^\star$ to denote an optimal policy, the corresponding optimal Q-function and optimal value function, respectively. We have the Bellman optimality equation

$$V^\star(s) = \max_{a \in \mathcal{A}} Q^\star(s, a), \quad Q^\star(s, a) = \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)}\big[r(s, a) + \gamma V^\star(s')\big]. \tag{3}$$

Meanwhile, the optimal policy $\pi^\star$ satisfies $\pi^\star(\cdot \,|\, s) = \operatorname{argmax}_\pi \langle Q^\star(s, \cdot), \pi(\cdot \,|\, s)\rangle_{\mathcal{A}}$. We aim to learn a policy π from the candidate policy class Π that maximizes the expected cumulative reward. Correspondingly, we define the performance metric as the sub-optimality compared with the optimal policy, i.e.,

$$\text{SubOpt}(\pi) = V^{\pi^\star}(s_0) - V^\pi(s_0). \tag{4}$$

2.1 BELLMAN CONSISTENT PESSIMISM

A unique challenge in offline RL is that the learned policy may induce a state-action density that is different from the data distribution μ, which may lead to large extrapolation errors when we do not impose any coverage assumption on μ. Therefore, it is important to carefully characterize the distribution shift, which we measure using the coverage coefficient. Specifically, we adopt the one used in Xie et al. (2021) that considers the distribution shift of Bellman errors:

Definition 1 (Bellman shift coefficient (Xie et al., 2021))**.** *We define $\mathcal{C}(\nu;\mu,\mathcal{Q},\pi)$ as follows to measure the distribution shift from an arbitrary distribution ν to the data distribution μ, w.r.t. $\mathcal{Q}$ and π,*

$$\mathcal{C}(\nu;\mu,\mathcal{Q},\pi) \coloneqq \max_{q\in\mathcal{Q}} \frac{\|q-\mathbb{T}^\pi q\|_{2,\nu}^2}{\|q-\mathbb{T}^\pi q\|_{2,\mu}^2}.$$

Here $\mathcal{Q}$ is the Q-function approximation class we consider. Intuitively, $\mathcal{C}(\nu;\mu,\mathcal{Q},\pi)$ measures how well Bellman errors under π transfer between the distributions ν and μ. For instance, a small value of $\mathcal{C}(d^\pi;\mu,\mathcal{Q},\pi)$ enables accurate policy evaluation for π using data collected under μ. Definition 1 is a generalization compared to prior works that is defined specific to linear function approximation (Agarwal et al., 2021; Jin et al., 2021). More generally, we have $\mathcal{C}(\nu;\mu,\mathcal{Q},\pi) \le \|\nu/\mu\|_\infty \coloneqq \sup_{s,a}\frac{\nu(s,a)}{\mu(s,a)}$ holds for any π and $\mathcal{Q}$.

2.2 Preference-Based Reinforcement Learning

To learn reward functions from preference labels, we consider the Bradley-Terry pairwise preference model (Bradley & Terry, 1952) as used by most prior works (Christiano et al., 2017; Ibarz et al., 2018; Palan et al., 2019). Specifically, the preference label between two given trajectories τ_i and τ_j is defined as

$$\mathbb{P}\left(\tau_i \succ \tau_j \,\middle|\, R\right) = \frac{1}{\exp\left(R(\tau_j)-R(\tau_i)\right)+1}, \tag{5}$$

where $\tau = (s_t,a_t)_{t=0}^T$ is a trajectory and $R(\tau)=\sum_{t=0}^T \gamma^t r(s_t,a_t)$ is the return function.

To simplify the theoretical analysis, we consider learning a *return* model instead of a *reward* model.

The return model $\widehat{R}$ is trained to minimize the cross-entropy loss between the predicted preference and the ground truth with a given preference dataset $\mathcal{D}_{\text{pref}}$ as follows:

$$\mathcal{L}_{\text{CE}}(R) = -\mathop{\mathbb{E}}_{(\tau^1,\tau^2,o)\sim\mathcal{D}_{\text{pref}}}\left[o\log\mathbb{P}\left(\tau_1\succ\tau_2\,\middle|\,R\right) + (1-o)\log(1-\mathbb{P}\left(\tau_1\succ\tau_2\,\middle|\,R\right))\right], \tag{6}$$

where o is the ground truth label given by human labelers.

We assume that the difference of return functions $\Delta\mathcal{R} \coloneqq \{\Delta R(\tau_1,\tau_2) : \text{Traj}\times\text{Traj}\to\mathbb{R} \mid \exists R\in\mathcal{R}, \Delta R(\tau_1,\tau_2) = R(\tau_1)-R(\tau_2)\}$ has a finite Eluder dimension, which is a common general function approximation class in RL literature (Russo & Van Roy, 2013; Chen et al., 2022).

Definition 2 (Eluder Dimension (Russo & Van Roy, 2013))**.** *Suppose $\mathcal{F}$ is a function class defined in $\mathcal{X}$, the α-Eluder dimension $d_{Elu}(\mathcal{F},\alpha)$ is the longest sequence $\{x_1,x_2,\cdots,x_n\}\in\mathcal{X}$ such that there exists $\alpha'\ge\alpha$ where x_i is α'-independent of $\{x_1,\cdots,x_{i-1}\}$ for all $i\in[n]$.*

The following generalized linear preference model considered by many prior works (Pacchiano et al., 2021; Zhan et al., 2023b) is a special case of finite Eluder dimension (Chen et al., 2022).

Definition 3 (Generalized Linear Preference Model)**.** *In d-dimensional generalized linear models, the preference function can be represented as $\mathbb{P}(\tau_1\succ\tau_2|\theta)=\sigma(\langle\phi(\tau_1,\tau_2),R\rangle)$ where σ is an increasing Lipschitz continuous function, $\phi:\text{Traj}\times\text{Traj}\to\mathbb{R}^d$ is a known feature map satisfying $\|\phi(\tau_1,\tau_2)\|_2\le H$ and $\theta\in\mathbb{R}^d$ is the unknown parameter.*

3 Method

In this section, we present our proposed algorithm, Offline Preference-based Reinforcement Learning with In-Dataset Exploration (OPRIDE), illustrated in Figure 1. The key idea of OPRIDE is to enhance the query efficiency of offline PbRL by conducting optimistic exploration with in-dataset queries and then utilizing the learned reward function pessimistically with discount factor scheduling.

Exploration is essential for gathering enough information about the optimal policy, while discount factor scheduling is crucial for mitigating the overoptimization of the learned reward function.

The overall algorithm is shown in Algorithm 1. In the sequel, we describe our method for query selection and utilization in detail.

3.1 Offline Query Selection with In-Dataset Exploration

Generating informative queries is crucial for calibrating the reward function. Various methods have been proposed to generate queries for offline preference-based RL, like disagreement-based approaches (Christiano et al., 2017) and information-gain-based approaches (Wilson et al., 2012; Shin et al., 2023), but they can still be inefficient in determining the optimal policy. This naturally leads to the idea of employing an exploration objective (Akrour et al., 2011) into offline query selection, where we maximize the information gain about the *optimal policy* rather than the *reward function*.

Inspired by principled exploration strategies for PbRL, analyzed in Section 4, we propose to use the difference of value differences as the exploration criteria. Specifically, we first train a set of reward functions $\{r_{\theta_i}\}_{i=1}^M$ using bootstapping, then train a set of value functions $\{V_{\psi_i}\}_{i=1}^M$ using offline algorithms like IQL (Kostrikov et al., 2021) with the reward functions. Finally, we select two trajectories τ_1 and τ_2 that maximize the difference of value differences between the two trajectories:

$$\underset{(\tau_1,\tau_2)\in\mathcal{D}}{\operatorname{argmax}}\ \underset{i,j\in[M]}{\operatorname{argmax}} \left|\left(V_{\psi_i}(\tau_1) - V_{\psi_j}(\tau_1)\right) - \left(V_{\psi_i}(\tau_2) - V_{\psi_j}(\tau_2)\right)\right|, \tag{7}$$

The reward function r_{θ_i} and the value functions Q_{ϕ_i}, V_{ψ_i} are iteratively updated after each preference query. The selection criteria in Equation 7 are rationalized by the theoretical analysis in Section 4. Intuitively, we should choose two trajectories τ_1, τ_2 such that there is a ψ_1 that strongly prefers τ_1 over τ_2, and there is a ψ_2 that strongly prefers τ_2 over τ_1. In such cases, we can obtain the maximum available information by acquiring the preference label between τ_1 and τ_2. Our proposed method has some distinct characteristics compared to previous methods. Compared to disagreement-based criteria (Shin et al., 2023), our method is scale-sensitive to the amount of difference, which gives us strong theoretical guarantees. Compared to variance-based methods, our method considers the difference in value functions instead of in reward functions. Compared to Bayesian methods (Lindner et al., 2021), our method uses critic values for query selection, ensuring easy implementation.

3.2 Policy Extraction with Variance-based Discount Scheduling

After obtaining the preference feedback, we can train the reward function using the cross-entropy loss in Equation 6 and annotate the reward-free dataset $\mathcal{D} = \{\{(s_t^n, a_t^n)\}_{t=0}^T\}_{n=1}^N$ to obtain a labeled dataset $\widehat{\mathcal{D}} = \{\{(s_t^n, a_t^n, \widehat{r}_t^n)\}_{t=0}^T\}_{n=1}^N$ where $\widehat{r} = 1/M \sum_{i=1}^M r_{\theta_i}$. However, it is well-known that a learned reward function is prone to overoptimization (Gao et al., 2023; Zhu et al., 2024), leading to overestimation of the value function and, subsequently, a suboptimal policy.

Learning from preference feedback is more vulnerable to this issue, as the feedback is binary and sparse. Empirically, we find that using a pessimistic ensemble of the reward function is insufficient to fully mitigate the overestimation issue in offline PbRL, as shown in Table 3. To solve this, we propose to adjust the discount factor based on the variance of the value function estimates that serve as a stronger regulator. Using a smaller discount factor is known to provide pessimistic and robust guarantees and performs well in various settings like imitation learning (Liu et al., 2024). Specifically, we reduce the discount factor where there is a higher variance in value estimation, thereby alleviating the impact of reward function overestimation.

$$\widehat{\gamma}(s,a) = \begin{cases} \gamma_{\text{small}}, & \text{if} \quad \text{Var}\{Q_{\phi_i}(s,a)\}_{i=1}^M > \text{Top } m\%(\{\text{Var}_j\{Q_{\phi_j}(s_j,a_j)\}_{i=1}^M\}_{j=1}^{|\text{Batch}|}) \\ \gamma, & \text{else} \end{cases} \tag{8}$$

where $\widehat{\gamma}$ is the adjusted discount factor. Please note that if the variance of the value estimation for a data point is greater than the top $m\%$ in the batch, we consider that the reward function for this data point has overestimation noise and reduces the corresponding discount factor.

Algorithm 1 Offline Preference-Based Reinforcement Learning with In-Dataset Exploration

---

1: **Input**: Unlabeled offline dataset $\mathcal{D} = \{\tau_n = \{(s_t^n, a_t^n)\}_{t=0}^T\}_{n=1}^N$, query budget K, ensemble number M
2: Initialized the preference dataset $\mathcal{D}_{\text{pref}} \leftarrow \varnothing$.
3: **for** episode $k = 1, \cdots, K$ **do**
4: Train M ensembles of reward network r_{θ_i} with $\mathcal{D}_{\text{pref}}$ using $\mathcal{L}_{\text{CE}}$ in Equation 6.
5: Train M corresponding value functions V_{ψ_i}, Q_{ϕ_i} with each reward function r_{θ_i} as in Equation 8.
6: Select trajectories $\tau^{k,1}, \tau^{k,2}$ that maximize the exploration objective according to Equation 7.
7: Receive the preference o_k between $\tau^{k,1}$ and $\tau^{k,2}$ and add it to the preference dataset, i.e.,

$$\mathcal{D}_{\text{pref}} \leftarrow \mathcal{D}_{\text{pref}} \cup \{(\tau^{k,1}, \tau^{k,2}, o_k)\}.$$

8: **end for**
9: Annotate the unlabeled offline dataset $\mathcal{D}$ with the reward function $\widehat{\theta}$ and obtain $\widehat{\mathcal{D}}$.
10: Adjust the discount facto γ to $\widehat{\gamma}$ based on Equation 8.
11: Extract policy π_ξ via Equation 9 from $\widehat{\mathcal{D}}$.
12: **Output**: The learned policy π_ξ

Subsequently, we can learn a corresponding Q-value function and extract the policy from the labeled datasets $\widehat{\mathcal{D}}$ by adopting the standard offline reinforcement learning algorithms, like IQL (Kostrikov et al., 2021):

$$\begin{aligned} L_V(\psi) &= \mathbb{E}_{(s,a)\sim\widehat{\mathcal{D}}}\left[L_2^\tau(Q_\phi(s,a) - V_\psi(s))\right] \\ &= \mathbb{E}_{(s,a)\sim\widehat{\mathcal{D}}}\left[|\tau - \mathbb{I}(Q_\phi(s,a) - V_\psi(s) < 0)|\right](Q_\phi(s,a) - V_\psi(s))^2 \\ L_Q(\phi) &= \mathbb{E}_{(s,a,\widehat{r})\sim\widehat{\mathcal{D}}}\left[\left(\widehat{r}(s,a) + \widehat{\gamma}(s,a)V_\psi(s') - Q_\phi(s,a)\right)^2\right] \\ L_\pi(\xi) &= \mathbb{E}_{(s,a)\sim\widehat{\mathcal{D}}}\left[\exp(\alpha A_{\psi,\phi}(s,a))\log(\pi_\xi(a|s))\right] \end{aligned}, \tag{9}$$

where π_ξ is extracted in a advantage-weighted manner and $A(s,a) = Q_\theta(s,a) - V_\psi(s)$ is the advantage function. $L_2^\tau(u) = |\tau - \mathbb{I}(u < 0)|u^2$ is the expectile regression loss, which is used to balance the conservatism and generalization in offline RL.

4 Theoretical Analysis

In this section, we investigate the theoretical guarantees for generating queries with an explorative objective. To simplify theoretical analysis, we consider the setting where we can make *online* queries along with access to a preference-free *offline* dataset. This is a good approximation when the available unsupervised trajectories for preference queries are abundant.

For a principled exploration strategy under such a setting, we can combine the wisdom from online PbRL and pessimistic value estimation for offline value estimation. Specifically, we consider the strategy to consist of the following procedures: (1) construct a confidence set for the return function based on existing queries; (2) construct a candidate policy set using pessimistic value estimation as the criteria; and (3) select a pair of policies that maximize disagreement on values for new queries. A detailed strategy description is available in Algorithm 2.

Construct Confidence Set. For the return function, we can use the cross entropy loss as in Equation 6 to get the maximum likelihood estimator (MLE) for the return function $\widehat{R}_k$. That is,

$$\widehat{R}_k = \operatorname*{argmin}_{R\in\mathcal{R}} L_k(R), \tag{10}$$

where $L_k(R) = \sum_{i=1}^k (o_i \log \mathbb{P}(\tau_i^1 \succ \tau_i^2; R) + (1 - o_i)\log(1 - \mathbb{P}(\tau_i^1 \succ \tau_i^2; R)))$ is the MLE loss. Then we can constuct the confidence set for the reward function as follows:

$$\mathcal{C}_k(\mathcal{R}) = \left\{ R \in \mathcal{R} \,\middle|\, \sum_{i=1}^{k} \left((R(\tau_i^1) - R(\tau_i^2)) - (\widehat{R}_k(\tau_i^1) - \widehat{R}_k(\tau_i^2)) \right)^2 \leq \beta_k \right\} \tag{11}$$

where β_k is the confidence parameter to be specified later.

Given the confidence set for the return function R, we can subsequently construct a confidence set for policies using a pessimistic value function. Specifically, we consider the pessimistic value function $\widehat{q}_R$ that leads to the worst-case value for the optimal policy over the Bellman uncertainty of the value function. Please refer to Algorithm 3 in Appendix A.1 for more details. The candidate policy set Π_k is constructed as follows:

$$\Pi_k = \left\{ \widehat{\pi} \mid \exists\, R \in \mathcal{C}_k(\mathcal{R}), \widehat{\pi} = \operatorname*{argmax}_{\pi \in \Pi} \widehat{q}_R(s_1, \pi). \right\}. \tag{12}$$

Intuitively speaking, Π_k consists of policies that are possibly optimal within the current level of uncertainty over reward and dynamics. By constraining exploration policies in Π_k, we achieve proper exploitation by avoiding unnecessary explorations.

Selecting Exploratory Policies. For a given pair of policies (π_1, π_2) in Π_k, we determine their exploration power by measuring how much disagreement can be made for different reward functions in the confidence set. Specifically, we select explorative policies via the following criteria:

$$\pi_1^k, \pi_2^k = \operatorname*{argmax}_{\pi_1, \pi_2 \in \Pi_k} \max_{R_1, R_2 \in \mathcal{C}_k(\mathcal{R})} \left(\left(\widehat{v}_{R_1}^{\pi_1} - \widehat{v}_{R_2}^{\pi_1} \right) - \left(\widehat{v}_{R_1}^{\pi_2} - \widehat{v}_{R_2}^{\pi_2} \right) \right). \tag{13}$$

Intuitively, we choose two policies π_1, π_2 such that there is a $R_1 \in \mathcal{C}_k(\mathcal{R})$ that strongly prefers π_1 over π_2, and there is a $R_2 \in \mathcal{C}_k(\mathcal{R})$ that strongly prefer π_2 over π_1.

Then we sample two trajectories $\tau^{k,1} \sim \pi^{k,1}, \tau^{k,2} \sim \pi^{k,2}$, query the preference between them, and add it to the preference dataset. Choosing the pair of trajectories that maximize disagreement helps us explore efficiently.

Theoretical Guarantees. We have the following theorem for our proposed Algorithm 2.

Theorem 4. *Let $\beta_k = c_1\sqrt{\log(K|\Delta\mathcal{R}|)/K}$ and $\epsilon = c_2\sqrt{\log(N|\Pi||\mathcal{Q}|)/N}$, where c_1, c_2 are universal constants. Then the expected suboptimality of $\bar{\pi}$ from Algorithm 2 is upper bounded by*

$$\mathrm{SubOpt}(\bar{\pi}) \leq \mathcal{O}\left(\underbrace{\sqrt{\frac{C^\dagger \log(N|\mathcal{Q}||\Pi|)}{N(1-\gamma)^2}}}_{\textit{Offline Error}} + \underbrace{\sqrt{\frac{\kappa d_{\textit{Elu}}(\Delta\mathcal{R}, 1/K) \log\left(K|\Delta\mathcal{R}|\right)}{K(1-\gamma)}}}_{\textit{Preference Error}} \right), \tag{14}$$

where κ is the degree of non-linearity of the link function σ, $C^\dagger$ is the coverage coefficient in Definition 1, N is the size of the offline dataset and K is the number of queries.

Proof. See Appendix B for a detailed proof. □

Equation 14 decomposes the suboptimality of Algorithm 2 into two terms nicely: the offline error term and the preference error term. The first error is due to the finite sample bias of the dataset, and the preference error is due to the limited amount of preference queries. Compared to pure online learning, the preference error is reduced by a factor of $1/(1-\gamma)$. Therefore, querying with an offline dataset can be much more sample-efficient than pure online queries when $N \gg K$. This is because the offline dataset contains rich information about dynamics and can reduce the effective horizon of the problem (Hu et al., 2023). This also aligns with our empirical findings that ~ 10 queries are usually sufficient for reasonable performance in offline settings.

Domain	Task	OPRL	PT	PT+PDS	IDRL	OPRIDE
Metaworld	lever-pull-v2	**63.2±10.4**	49.2±3.7	51.7±0.1	33.1±1.2	51.8±1.6
	peg-insert-side	3.5±1.8	16.8±0.1	12.4±1.4	67.4±0.1	**79.0±0.2**
	plate-slide	77.4±1.6	4.9±0.0	37.3±2.3	**79.6±3.5**	**79.9±4.6**
	push	10.6±1.5	16.7±5.0	1.8±0.4	30.7±5.3	**39.3±3.4**
	push-back	0.8±0.0	1.1±0.4	1.1±0.1	14.0±1.1	**17.7±2.0**
	push-wall	7.4±4.2	74.8±14.4	3.4±0.9	89.2±3.2	**102.2±1.2**
	reach	63.5±2.9	82.0±0.8	84.3±0.9	75.8±1.8	**88.0±0.5**
	soccer	34.3±4.0	**51.3±4.1**	41.5±11.9	44.3±2.1	45.4±3.9
	sweep-into	37.1±13.9	9.8±0.2	9.2±0.1	63.1±3.5	**71.6±0.1**
	sweep	6.8±1.8	8.0±0.4	8.0±0.1	73.0±2.8	**78.5±1.0**
Average		30.4±4.2	31.4±2.9	25.0±1.8	57.0±6.3	**65.3±3.3**

Table 1: Performance of offline RL algorithm on the reward-labeled dataset with different preference reward learning methods on the Meta-World tasks. All experiment results were averaged over five random seeds. Please refer to Appendix D for the complete experimental results.

Domain	Task	OPRL	PT	PT+PDS	IDRL	OPRIDE
Antmaze	umaze	76.3±3.7	77.5±4.5	84.5±8.5	85.5±3.4	**87.5±5.6**
	umaze-diverse	72.5±3.4	68.0±3.0	**78.0±6.0**	69.1±4.2	73.1±2.4
	medium-play	0.0±0.0	63.5±0.5	**72.5±6.5**	63.8±4.1	62.2±2.0
	medium-diverse	0.0±0.0	63.5±4.5	58.0±4.0	65.7±4.1	**69.4±5.2**
	large-play	7.3±0.9	6.5±2.5	9.0±8.0	18.7±3.4	**27.5±12.5**
	large-diverse	6.9±2.4	**23.5±0.5**	8.5±2.5	14.3±2.5	21.5±1.5
Average		27.1±1.7	50.4±2.5	51.7±5.9	52.8±3.6	**56.8±4.8**

Table 2: Performance of offline RL algorithm on the reward-labeled dataset with different preference reward learning methods on the Antmaze tasks.

5 EXPERIMENTS

In this section, we aim to answer the following questions: (1) How does our method perform on various navigation and manipulation tasks compared to other offline PbRL methods? (2) How effective is the proposed exploration-based query selection and discounted-based pessimism? (3) How does our method perform across different numbers of queries?

5.1 EXPERIMENTAL DETAILS

Environment Setup. We perform empirical evaluations on Meta-World (Yu et al., 2019) and the Antmaze task on the D4RL benchmark (Fu et al., 2020). In the preference query, we use a segment length of 50 for all tasks. We adopt the normalized score metric proposed by the D4RL benchmark, averaging over five random seeds with standard deviation. Scores roughly range from 0 to 100, where 0 corresponds to the performance of a random policy, and 100 indicates the performance of an expert. Please refer to Appendix E for more experimental details.

Baselines. Offline Preference-based Reinforcement Learning (OPRL; Shin et al., 2023) is a representative algorithm in offline PbRL, which proposes various mechanisms to select queries (e.g., disagreement technique). Recently, Preference Transformer (PT; Kim et al., 2023) achieved state-of-the-art performance by using a transformer architecture to model the potential non-Markovian reward function. Our method adopts the same architecture as in Preference Transformer (PT). To illustrate the effectiveness of our proposed variance-based discount, we compare our method with Provable Data Sharing (PDS; Hu et al., 2023) as a baseline algorithm, which proposes to use a pessimistic ensemble to account for uncertainties in the reward function, thus reducing the potential overoptimization issues in the learned reward function. We adopt the same architecture as in Preference Transformer (PT) for a fair comparison. In addition, we also adopt IDRL (Lindner et al., 2021) as our baseline, which

Task	PT	PDS + Random Query	VDS + Random Query	VDS + Disagreement	OPRIDE (VDS+IDE)
bin-picking	31.9±16.2	53.4±19.0	71.9±9.0	78.5±17.8	**93.3**±**3.2**
button-press-wall	58.8±0.9	59.4±0.9	**77.2**±**0.8**	67.4±5.4	**77.7**±**0.1**
door-close	65.1±10.1	62.4±8.7	72.3±0.1	88.3±0.7	**94.8**±**1.1**
faucet-close	57.8±0.9	46.2±0.2	59.4±8.5	48.7±0.6	**73.1**±**0.8**
peg-insert-side	16.8±0.1	12.4±1.4	13.8±4.4	9.7±8.5	**79.0**±**0.2**
reach	82.0±0.8	84.3±0.9	83.3±0.1	**86.6**±**0.1**	**88.0**±**0.5**
sweep	8.0±0.4	8.0±0.1	28.7±1.8	18.2±2.9	**78.5**±**1.0**

Table 3: Ablation of the query selection module on the Meta-World tasks. We report the performance of offline RL algorithm on the reward-labeled dataset with various query selection and policy extration mechanism. IDE and VDS represent the In-Dataset Exploration module and the Variance-based Discount Scheduling module proposed in Section 3.1, respectively.

Domain	Tasks	Zero	Random	Negative	OPRIDE
Metaworld	coffee-push	7.6±4.3	5.8±2.7	0.7±0.1	**59.4**±**24.8**
	disassemble	9.3±0.4	**16.8**±**7.3**	10.1±0.2	12.4±2.9
	hammer	38.1±6.4	**46.1**±**2.4**	22.6±1.8	39.2±11.2
	push	**57.5**±**1.5**	34.4±17.3	4.6±2.3	39.3±20.4
	push-wall	81.9±3.8	80.1±0.9	17.6±1.9	**102.2**±**1.2**
	soccer	33.3±1.6	41.1±8.8	44.0±6.4	**45.4**±**3.9**
	sweep	29.0±0.2	29.0±2.6	24.9±0.3	**78.5**±**1.0**

Table 4: Comparison between the survival instinct and OPRIDE.

proposes an information-directed query selection method and uses the Laplacian approximation and the Hessian matrix for posterior computation.

5.2 Experimental Results

Answer to Question 1: To show that OPRIDE can generate valuable rewards with a few queries, we conducted a comprehensive comparative analysis of OPRIDE against several baseline methods, utilizing Meta-World and Antmaze tasks as our testing grounds. Specifically, we use a budget of 10 queries on each task for all offline preference-based reinforcement learning methods. Then, we let all algorithms employ the IQL algorithm for subsequent offline training for a fair comparison. The experimental results in Table 1 and Table 2 are normalized episode returns averaged over five random seeds. In 22 out of 30 tasks in Meta-World and Antmaze, OPRIDE demonstrates superior performance compared to baseline algorithms. Moreover, unlike IDRL, which relies on the Laplacian approximation and the Hessian matrix for posterior computation, our method leverages critic values for query selection, ensuring easier implementation and superior empirical performance, as demonstrated in our comparative experiments.

We also compare OPRIDE with the recent research work Survival Instinct (Li et al., 2024) since they find that wrong rewards can also lead to good offline RL performance. Specifically, we used three types of rewards, the same as the author: (1) zero: the zero reward, (2) random: labeling each transition with a reward value randomly sampled from Unif [0, 1], and (3) negative: the negation of true reward. Then, we trained the same offline learning algorithm as OPRIDE on the reward-labeled dataset. The experimental results in Table 4 indicate that OPRIDE still outperforms these baselines in most tasks. We attribute the above experimental results to the challenging nature of the dataset we created. Specifically, in Li et al. (2024), the perturbed script policy data accounts for 100% of the dataset. However, in our created dataset, the perturbed script policy data only accounts for 5% of the dataset. We conduct additional experiments on Mujoco and Kitchen tasks. Please refer to Appendix D for the complete experimental results.

Answer to Question 2: To study the contribution of each component in our framework, we conduct several ablation studies to verify the effectiveness of each part, as shown in Table 3.

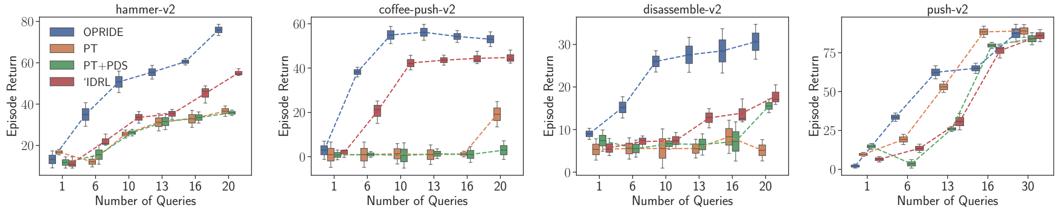

Figure 2: Performance of offline preference-based RL algorithms with various queries. OPRIDE achieves a better query efficiency across tasks and number of queries.

γ_{small}	0.5	0.6	0.7	0.8	0.9	0.95
bin-picking	72.1±23.9	87.8±2.7	93.3±3.2	84.6±9.4	74.8±33.5	70.9±9.4
button-press-wall	77.6±0.3	77.4±0.3	77.7±0.1	71.0±0.7	69.2±0.9	68.1±9.7
door-close	88.4±0.8	89.9±0.7	94.8±1.1	90.0±2.1	91.1±1.5	87.6±0.7
faucet-close	58.1±5.2	58.2±12.5	73.1±0.8	61.4±2.7	55.1±3.7	57.4±12.1

Table 5: Performance of offline RL algorithm on the reward-labeled dataset with various discount factor values γ_{small} on the high variance data points.

Comparing our method with the `VDS + Random Query` and the `VDS + Disagreement` baseline, we can see that disagreement-based approaches offer little improvement over the random query selection baseline, while our exploration criteria lead to vast performance improvement, showcasing that our method is able to collect useful information within a few queries. Comparing the `PDS + Random Query` and the `VDS + Random Query` baseline, we can conclude that while PDS is helpful on some tasks like `bin-picking-v2`, it fails to prevent reward overoptimization and makes the performance worse on some other tasks. On the contrary, `VDS + Random Query` is able to improve over the `PT` baseline on most tasks, showing its robust ability to reduce reward overestimation. Overall, our method achieves the best performance compared to other ablation baselines, demonstrating the effectiveness of each part of our algorithm.

We have conducted ablation experiments to determine the sensitivity of the discount factor hyper-parameter. Specifically, we vary the γ_{small} values from 0.5 to 0.95 for the data points with the high variance. The experimental results in Table 5 indicate that 0.7∼0.8 is an appropriate range for γ_{small}, and the performance is robust across different γ_{small} values. We conduct additional ablation studies for the In-Dataset Exploration module and the Variance-based Discount Scheduling module. Please refer to Appendix D for the complete experimental results.

Answer to Question 3: To investigate how the number of queries affects OPRIDE 's overall performance, we vary the number of queries and compare our method with various baselines. The results presented in Figure 2 demonstrate that OPRIDE achieves a superior query efficiency and significantly outperforms the baselines across various numbers of queries. In most tasks, OPRIDE achieves good performance with just ten queries, and its performance continues to improve as the number of queries increases. In contrast, the baseline methods require multiple times the number of queries to achieve performance on par with OPRIDE (e.g., `hammer-v2`). Even with 20 queries, the baseline algorithm shows no significant improvement on some hard tasks (e.g., `coffee-push-v2`).

6 CONCLUSION

This paper proposes a new framework, in-dataset exploration, to improve query efficiency in offline PbRL. Compared with disagreement-based approaches, using an exploration strategy helps reduce the burden of learning an accurate reward function in the low-return region, improving learning efficiency. Our proposed algorithm, OPRIDE, conducts in-dataset exploration by weighted trajectory queries, and a principled exploration strategy deals with pairwise queries. Our method has provable guarantees, and our practical variant achieves strong empirical performance on various tasks. Compared to prior methods, our method significantly reduces the required queries. Overall, our method provides a promising and principled way to reduce queries required from human labelers in PbRL.

REFERENCES

Agarwal, A., Kakade, S., Krishnamurthy, A., and Sun, W. Flambe: Structural complexity and representation learning of low rank mdps. *Advances in neural information processing systems*, 33: 20095–20107, 2020.

Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98): 1–76, 2021.

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Ajay, A., Kumar, A., Agrawal, P., Levine, S., and Nachum, O. Opal: Offline primitive discovery for accelerating offline reinforcement learning. *arXiv preprint arXiv:2010.13611*, 2020.

Akrour, R., Schoenauer, M., and Sebag, M. Preference-based policy learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011. Proceedings, Part I 11*, pp. 12–27. Springer, 2011.

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

Anthropic. Introducing claude, 2023. URL `https://www.anthropic.com/news/introducing-claude`.

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

Cai, Q., Yang, Z., Jin, C., and Wang, Z. Provably efficient exploration in policy optimization. In *International Conference on Machine Learning*, pp. 1283–1294. PMLR, 2020.

Chen, X., Zhong, H., Yang, Z., Wang, Z., and Wang, L. Human-in-the-loop: Provably efficient preference-based reinforcement learning with general function approximation. In *International Conference on Machine Learning*, pp. 3773–3793. PMLR, 2022.

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

Degrave, J., Felici, F., Buchli, J., Neunert, M., Tracey, B., Carpanese, F., Ewalds, T., Hafner, R., Abdolmaleki, A., de Las Casas, D., et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

Ghosh, D., Bhateja, C., and Levine, S. Reinforcement learning from passive data via latent intentions. *arXiv preprint arXiv:2304.04782*, 2023.

Hu, H., Yang, Y., Zhao, Q., and Zhang, C. On the role of discount factor in offline reinforcement learning. In *International Conference on Machine Learning*, pp. 9072–9098. PMLR, 2022.

Hu, H., Yang, Y., Zhao, Q., and Zhang, C. The provable benefits of unsupervised data sharing for offline reinforcement learning. *arXiv preprint arXiv:2302.13493*, 2023.

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

Jiang, N., Kulesza, A., Singh, S., and Lewis, R. The dependence of effective planning horizon on model accuracy. In *Proceedings of the 2015 international conference on autonomous agents and multiagent systems*, pp. 1181–1189, 2015.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is q-learning provably efficient? *Advances in neural information processing systems*, 31, 2018.

Jin, Y., Yang, Z., and Wang, Z. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pp. 5084–5096. PMLR, 2021.

Kang, Y., Shi, D., Liu, J., He, L., and Wang, D. Beyond reward: Offline preference-guided policy optimization. *arXiv preprint arXiv:2305.16217*, 2023.

Kim, C., Park, J., Shin, J., Lee, H., Abbeel, P., and Lee, K. Preference transformer: Modeling human preferences using transformers for rl. *arXiv preprint arXiv:2303.00957*, 2023.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

Lee, K., Smith, L. M., and Abbeel, P. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6152–6163. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/lee21i.html.

Li, A., Misra, D., Kolobov, A., and Cheng, C.-A. Survival instinct in offline reinforcement learning. *Advances in neural information processing systems*, 36, 2024.

Lindner, D., Turchetta, M., Tschiatschek, S., Ciosek, K., and Krause, A. Information directed reward learning for reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 3850–3862, 2021.

Liu, Y., Dong, W., Hu, Y., Wen, C., Yin, Z.-H., Zhang, C., and Gao, Y. Imitation learning from observation with automatic discount scheduling. *arXiv preprint arXiv:2310.07433*, 2023.

Liu, Y., Dong, W., Hu, Y., Wen, C., Yin, Z.-H., Zhang, C., and Gao, Y. Imitation learning from observation with automatic discount scheduling, 2024.

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

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

Pacchiano, A., Saha, A., and Lee, J. Dueling rl: Reinforcement learning with trajectory preferences. *arXiv preprint arXiv:2111.04850*, 2021.

Palan, M., Landolfi, N. C., Shevchuk, G., and Sadigh, D. Learning reward functions by integrating human demonstrations and preferences. *arXiv preprint arXiv:1906.08928*, 2019.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.

Rafailov, R., Hejna, J., Park, R., and Finn, C. From r to q^*: Your language model is secretly a q-function, 2024.

Russo, D. and Van Roy, B. Eluder dimension and the sample complexity of optimistic exploration. In *NIPS*, pp. 2256–2264. Citeseer, 2013.

Russo, D. and Van Roy, B. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

Sekhari, A., Sridharan, K., Sun, W., and Wu, R. Contextual bandits and imitation learning via preference-based active queries. *arXiv preprint arXiv:2307.12926*, 2023.

Shin, D., Dragan, A. D., and Brown, D. S. Benchmarks and algorithms for offline preference-based reward learning. *arXiv preprint arXiv:2301.01392*, 2023.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 3008–3021. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf.

Team, O. Gpt-4 technical report, 2024.

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

Wang, Y., Liu, Q., and Jin, C. Is rlhf more difficult than standard rl? *arXiv preprint arXiv:2306.14111*, 2023.

Wilson, A., Fern, A., and Tadepalli, P. A bayesian approach for policy learning from trajectory preference queries. *Advances in neural information processing systems*, 25, 2012.

Wu, R. and Sun, W. Making rl with preference-based feedback efficient via randomization. *arXiv preprint arXiv:2310.14554*, 2023.

Xie, T., Cheng, C.-A., Jiang, N., Mineiro, P., and Agarwal, A. Bellman-consistent pessimism for offline reinforcement learning. *Advances in neural information processing systems*, 34:6683–6694, 2021.

Xu, Y., Wang, R., Yang, L., Singh, A., and Dubrawski, A. Preference-based reinforcement learning with finite-time guarantees. *Advances in Neural Information Processing Systems*, 33:18784–18794, 2020.

Yang, Y., Hu, H., Li, W., Li, S., Yang, J., Zhao, Q., and Zhang, C. Flow to control: Offline reinforcement learning with lossless primitive discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10843–10851, 2023.

Ye, W., Zhang, Y., Abbeel, P., and Gao, Y. Become a proficient player with limited data through watching pure videos. In *The Eleventh International Conference on Learning Representations*.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2019. URL https://arxiv.org/abs/1910.10897.

Yu, T., Kumar, A., Chebotar, Y., Hausman, K., Finn, C., and Levine, S. How to leverage unlabeled data in offline reinforcement learning. In *International Conference on Machine Learning*, pp. 25611–25635. PMLR, 2022.

Zhan, W., Uehara, M., Kallus, N., Lee, J. D., and Sun, W. Provable offline reinforcement learning with human feedback. *arXiv preprint arXiv:2305.14816*, 2023a.

Zhan, W., Uehara, M., Sun, W., and Lee, J. D. How to query human feedback efficiently in rl? *arXiv preprint arXiv:2305.18505*, 2023b.

Zhu, B., Jordan, M. I., and Jiao, J. Iterative data smoothing: Mitigating reward overfitting and overoptimization in rlhf. *arXiv preprint arXiv:2401.16335*, 2024.

A Additional Details

In this section, we provide a detailed description for the theoretical version of OPRIDE as in Algorithm 2.

Algorithm 2 OPRIDE, theoretical version

1: **Input**: Unlabeled offline dataset $\mathcal{D}$, query budget K
2: Initialized preference dataset $\mathcal{D}_{\text{pref}} \leftarrow \emptyset$.
3: **for** $k = 1, \cdots, K$ **do**
4: Calculate confidence set $\mathcal{C}_k(\mathcal{R})$ for reward function based on $\mathcal{D}_{\text{pref}}$ with Equation 11.
5: Calculate pessimistic value function $\widehat{q}(\cdot)$ using Algorithm 3 for each reward function in $\mathcal{C}_k(\mathcal{R})$.
6: Construct the near-optimal policy set Π_k using Equation 12.
7: Select explorative policies π_k^1, π_k^2 within Π_k based on Equation 13.
8: Sample trajectories τ_k^1, τ_k^2 with selected policy π_k^1, π_k^2.
9: Receive the preference o_k between τ_k^1 and τ_k^2 and add it to the preference dataset

$$\mathcal{D}_{\text{pref}} \leftarrow \mathcal{D}_{\text{pref}} \cup \{(\tau_k^1, \tau_k^2, o_k)\}.$$

10: **end for**
11: **Output**: Average policy $\bar{\pi} = \frac{1}{2K} \cdot \sum_{k=1}^{K} (\pi_k^1 + \pi_k^2)$.

A.1 Details of Bellman-consistent Pessimism (BCP; Xie et al., 2021)

In this section, we consider *Bellman-consistent Pessimism* (BCP; Xie et al., 2021) as the backbone algorithm, described in Algorithm 3. It is a representative model-free offline algorithm with theoretical guarantees. PEVI uses negative bonus $\Gamma(\cdot, \cdot)$ over standard Q-value estimation $\widehat{Q}(\cdot, \cdot) = (\widehat{\mathbb{B}}\widehat{V})(\cdot)$ to reduce potential bias due to finite data, where $\widehat{\mathbb{B}}$ is some empirical estimation of $\mathbb{B}$ from dataset $\mathcal{D}$. We use the following notion of ξ-uncertainty quantifier as follows to formalize the idea of pessimism.

Algorithm 3 Bellman-consistent Pessimism (BCP)

1: **Input**: Offline Dataset $\mathcal{D}_{\text{off}} = \{\tau_k = \{(s_t^k, a_t^k)\}_{t=0}^{T}\}_{k=1}^{K}$, reward function r.
2: Set the loss function as

$$\mathcal{L}(q, q', \pi; \mathcal{D}) = \sum_{k=1}^{K} \sum_{t=0}^{T} \left(q_t(s_t^k, a_t^k) - (r(s_t^k, a_t^k) + \gamma q'_{t+1}(s_{t+1}^k, \pi_{t+1})) \right)^2. \quad (15)$$

3: Set the confidence set of value functions as

$$\mathcal{V}(\pi, \epsilon) = \left\{ q \in \mathcal{V} : \mathcal{L}(q, q, \pi; \mathcal{D}) - \min_{q' \in \mathcal{V}} \mathcal{L}(q', q, \pi; \mathcal{D}) \le \epsilon \right\}. \quad (16)$$

4: Compute pessimistic policy and value function as

$$\widehat{\pi} = \underset{\pi \in \Pi}{\text{argmax}} \min_{q \in \mathcal{V}(\pi, \epsilon)} q_1(s_1, \pi). \quad (17)$$

and

$$\widehat{q} = \underset{q \in \mathcal{V}(\widehat{\pi}, \epsilon)}{\text{argmin}}\, q_1(s_1, \widehat{\pi}). \quad (18)$$

5: **Output**: $\widehat{\pi}$ and $\widehat{q}$.

Algorithm 4 Bellman-consistent Pessimism Evaluation

1: **Input**: Offline Dataset $\mathcal{D}_{\text{off}} = \{\tau_k = \{(s_t^k, a_t^k)\}_{t=0}^T\}_{k=1}^K$, reward function r, policy π
2: Set the loss function as

$$\mathcal{L}(q, q', \pi; \mathcal{D}) = \sum_{k=1}^{K} \sum_{t=0}^{T} \left(q_t(s_t^k, a_h^k) - (r(s_t^k, a_t^k) + \gamma q'(s_{t+1}^k, \pi_{t+1})) \right)^2 . \tag{19}$$

3: Set the confidence set of value functions as

$$\mathcal{V}(\pi, \epsilon) = \left\{ q \in \mathcal{V} : \mathcal{L}(q, q, \pi; \mathcal{D}) - \min_{q' \in \mathcal{V}} \mathcal{L}(q', q, \pi; \mathcal{D}) \leq \epsilon \right\}. \tag{20}$$

4: Compute pessimistic value function as

$$\widehat{q} = \operatorname*{argmin}_{q \in \mathcal{V}(\pi, \epsilon)} q(s_0, \pi). \tag{21}$$

5: **Output**: $\widehat{v}$ and $\widehat{q}$.

Lemma 5. *Under conditions of Theorem, let $C^\dagger = \mathcal{C}(d_{\pi^\star}; \mu, \mathcal{Q}, \pi^\star)$, we have*

$$V^\star(\pi^\star) - V^\star(\widehat{\pi}) \leq O\left(\sqrt{\frac{C^\dagger \log \frac{|\mathcal{Q}||\Pi|}{\delta}}{N(1-\gamma)^2}} \right), \tag{22}$$

where $\widehat{\pi}$ is the output of Algorithm 3 with dataset $\mathcal{D}_{off}$ and return function R. Similarly, we have

$$V^\star(\pi) - \widehat{v}(\pi) \leq O\left(\sqrt{\frac{C^\dagger \log \frac{|\mathcal{Q}||\Pi|}{\delta}}{N(1-\gamma)^2}} \right), \tag{23}$$

where $\widehat{v}$ is the output of Algorithm 4 with dataset $\mathcal{D}_{off}$, policy π and return function R.

Proof. This proof is mainly adapted from the proof of Theorem 1 in Xie et al. (2021) to the finite-horizon case. For simplicity we only prove the first part of the lemma. The second part can be proved similarly using the pessimistic property of the value function $\widehat{v}$.

Using the optimality of $\widehat{\pi}$, we have

$$\max_{v \in \mathcal{Q}_{\pi, \epsilon_r}} v(s_0, \pi) - \min_{v \in \mathcal{Q}_{\widehat{\pi}, \epsilon_r}} v(s_0, \widehat{\pi}) \leq \max_{v \in \mathcal{Q}_{\pi, \epsilon_r}} v(s_0, \pi) - \min_{v \in \mathcal{Q}_{\pi, \epsilon_r}} v(s_0, \pi).$$

Now, let $v_{\min}(\pi) := \operatorname{argmin}_{v \in \mathcal{Q}_{\pi, \epsilon_r}} v(s_0, \pi)$ and $v_{\max}(\pi) := \operatorname{argmax}_{v \in \mathcal{Q}_{\pi, \epsilon_r}} v(s_0, \pi)$.

Using a standard reward decomposition argument Cai et al. (2020), we have

$$\begin{aligned}
& v_{1,max}(\pi) - v_{1,min}(\pi) \\
&= v_{1,max} - v_1(\pi) + v_1(\pi) - v_{1,min} \\
&= \mathbb{E}_{d_\pi} \left[\sum_{h=1}^{H} (v_{h,max} - \mathbb{T}^\pi v_{h+1,max}) - \sum_{h=1}^{H} (v_{h,min} - \mathbb{T}^\pi v_{h+1,min}) \right] \\
&\leq \sum_{h=1}^{H} \|v_{h,max} - \mathbb{T}^\pi v_{h+1,max}\|_{2,d^\pi} + \|v_{h,min} - \mathbb{T}^\pi v_{h+1,min}\|_{2,d^\pi} \\
&\leq \sqrt{\mathcal{C}(d^\pi; \mu, \mathcal{V}, \pi)} \sum_{i=1}^{H} (\|v_{h,max} - \mathbb{T}^\pi v_{h+1,max}\|_{2,\mu} + \|v_{h,min} - \mathbb{T}^\pi v_{h+1,min}\|_{2,\mu}) \\
&\leq \frac{1}{1-\gamma} \sqrt{\mathcal{C}(d^\pi; \mu, \mathcal{V}, \pi) \epsilon_b}, \qquad (24)
\end{aligned}$$

(25)

holds under event $\mathcal{E}_2$ in Lemma 12 and $\mathcal{E}_3$ in Lemma 13. The second inequality follows from the definition of $\mathcal{C}(d^\pi; \mu, \mathcal{V}, \pi)$ and the last inequality follows from Lemma 12 and Lemma 13. Let $\pi = \pi^\star$ and plug in the definition of ϵ_b, we complete the proof.

□

B Proof of Theorem 4

Theorem 6 (Restatement of Theorem 4)**.** *Suppose (1) $Q^\star \in \mathcal{Q}, \pi^\star \in \Pi$, and (2) $\mathbb{T}^\pi q \in \mathcal{Q}, \forall \pi \in \Pi, q \in \mathcal{Q}$. Also suppose the difference of return functions has a finite Eluder dimension $d_{Elu}(\Delta R, \alpha)$ and the underlying distribution of the offline dataset admit a finite coverage coefficient $C^\dagger$. Let $\beta_k = c_1\sqrt{\log(K|\Delta\mathcal{R}|)/K}$ and $\epsilon = c_2\sqrt{\log(N|\Pi||\mathcal{Q}|)/N}$, where c_1, c_2 are universal constants. Then the expected suboptimality of $\bar{\pi}$ from Algorithm 2 is upper bounded by*

$$\text{SubOpt}(\bar{\pi}) \le \mathcal{O}\left(\sqrt{\frac{C^\dagger \log(N|\mathcal{Q}||\Pi|)}{N(1-\gamma)^2}} + \sqrt{\frac{d_{Elu}(\Delta\mathcal{R}, 1/K)\log(K|\Delta\mathcal{R}|)}{K(1-\gamma)}}\right), \tag{26}$$

where N is the size of the offline dataset and K is the number of queries.

Remark 7. *In Theorem 6 we consider finite function classes for policy Π, Q-value $\mathcal{Q}$ and return function $\mathcal{R}$. However, it can be readily extended to infinite function classes by using the covering number of the function classes, as done in Chen et al. (2022); Xie et al. (2021). We also remark that while we consider the realizable and Bellman-complete setting where $Q^\star \in \mathcal{Q}$ and $\mathbb{T}Q \in \mathcal{Q}$ for simplicity, we can extend the result to approximate realizable and Bellman-complete setting as in Xie et al. (2021).*

Remark 8. *The suboptimality bound uses the Eluder dimension of the difference function class $\Delta\mathcal{R}$ of the original return function class $\mathcal{R}$. This is because we can only determine $R(\tau_1) - R(\tau_2)$ from the preference query between τ_1 and τ_2 and the absolute value for $R(\tau)$ can be free to choose.*

Proof. For simplicity we let $V^\pi := V_1^\pi(s_1)$.

For any return function $\widetilde{R} \in \mathcal{C}_k(\mathcal{R})$ and the policy $\widetilde{\pi} = \text{BCP}(\mathcal{D}, \widetilde{R})$ generated by Algorithm 3, we have

$$\begin{aligned}
&V_{R^\star}^{\pi^\star} - V_{R^\star}^{\widetilde{\pi}} \qquad (27)\\
=&V_{R^\star}^{\pi^\star} - \widehat{v}_{R^\star}^{\pi^\star} + \widehat{v}_{R^\star}^{\pi^\star} - \widehat{v}_{\widetilde{R}}^{\pi^\star} + \widehat{v}_{\widetilde{R}}^{\pi^\star} - \widehat{v}_{\widetilde{R}}^{\widetilde{\pi}} + \widehat{v}_{\widetilde{R}}^{\widetilde{\pi}} - \widehat{v}_{R^\star}^{\widetilde{\pi}} + \widehat{v}_{R^\star}^{\widetilde{\pi}} - V_{R^\star}^{\widetilde{\pi}}\\
\le&V_{R^\star}^{\pi^\star} - \widehat{v}_{R^\star}^{\pi^\star} + \widehat{v}_{R^\star}^{\pi^\star} - \widehat{v}_{\widetilde{R}}^{\pi^\star} + \widehat{v}_{\widetilde{R}}^{\pi^\star} - \widehat{v}_{\widetilde{R}}^{\widetilde{\pi}} + \widehat{v}_{\widetilde{R}}^{\widetilde{\pi}} - \widehat{v}_{R^\star}^{\widetilde{\pi}} + 0\\
\le&V_{R^\star}^{\pi^\star} - \widehat{v}_{R^\star}^{\pi^\star} + \widehat{v}_{R^\star}^{\pi^\star} - \widehat{v}_{\widetilde{R}}^{\pi^\star} + 0 \qquad + \widehat{v}_{\widetilde{R}}^{\widetilde{\pi}} - \widehat{v}_{R^\star}^{\widetilde{\pi}}\\
\le&V_{R^\star}^{\pi^\star} - \widehat{v}_{R^\star}^{\pi^\star} + \max_{R_1, R_2 \in \mathcal{C}_k(\mathcal{R})}\left(\widehat{v}_{R_1}^{\pi^\star} - \widehat{v}_{R_2}^{\pi^\star} + \widehat{v}_{R_2}^{\widetilde{\pi}} - \widehat{v}_{R_1}^{\widetilde{\pi}}\right)\\
\le&V_{R^\star}^{\pi^\star} - \widehat{v}_{R^\star}^{\pi^\star} + \max_{R_1, R_2 \in \mathcal{C}_k(\mathcal{R})}\left(\widehat{v}_{R_1}^{\widetilde{\pi}^{k,1}} - \widehat{v}_{R_2}^{\widetilde{\pi}^{k,1}} + \widehat{v}_{R_2}^{\widetilde{\pi}^{k,2}} - \widehat{v}_{R_1}^{\widetilde{\pi}^{k,2}}\right), \qquad (28)
\end{aligned}$$

which hold under event $\mathcal{E}_1$ in Lemma 9. The first inequality follows from the pessimistic property of $\widehat{v}$, the second inequality follows from the fact that $\widetilde{\pi}$ is the optimal policy with respect to $\widehat{v}_{\widetilde{R}}$. The third inequality holds since $\widetilde{R}, R^\star \in \mathcal{C}_k(\mathcal{R})$ and the last inequality follows from the definition of $\widetilde{\pi}^{k,1}, \widetilde{\pi}^{k,2}$.

Following Lemma 5, we have for all policy π and reward function R, the following holds with probability at least $1 - 2\delta$:

$$|V_R^\pi - \widehat{v}_R^\pi| \le c \cdot \sqrt{\frac{C^\dagger \log(N|\Pi||\mathcal{Q}|)}{N(1-\gamma)^2}} := \mathcal{E}_{\text{off}}.$$

Then we have

$$
\begin{aligned}
& V^{\pi^\star}_{R^\star} - V^{\widetilde{\pi}}_{R^\star} && (29)\\
&\le V^{\pi^\star}_{R^\star} - \widehat{v}^{\pi^\star}_{R^\star} + \max_{R_1, R_2 \in \mathcal{C}_k(\mathcal{R})} \left(\widehat{v}^{\widetilde{\pi}^{k,1}}_{R_1} - \widehat{v}^{\widetilde{\pi}^{k,1}}_{R_2} + \widehat{v}^{\widetilde{\pi}^{k,2}}_{R_2} - \widehat{v}^{\widetilde{\pi}^{k,2}}_{R_1} \right) \\
&\le \mathcal{E}_{\text{off}} + \max_{R_1, R_2 \in \mathcal{C}_k(\mathcal{R})} \left(\left(V^{\widetilde{\pi}^{k,1}}_{R_1} - V^{\widetilde{\pi}^{k,1}}_{R_2} + V^{\widetilde{\pi}^{k,2}}_{R_2} - V^{\widetilde{\pi}^{k,2}}_{R_1} \right) + \left(\widehat{v}^{\widetilde{\pi}^{k,1}}_{R_1} - V^{\widetilde{\pi}^{k,1}}_{R_1} \right) \right. \\
&\quad \left. + \left(\widehat{v}^{\widetilde{\pi}^{k,1}}_{R_2} - V^{\widetilde{\pi}^{k,1}}_{R_2} \right) + \left(\widehat{v}^{\widetilde{\pi}^{k,2}}_{R_2} - V^{\widetilde{\pi}^{k,2}}_{R_2} \right) + \left(\widehat{v}^{\widetilde{\pi}^{k,2}}_{R_1} - V^{\widetilde{\pi}^{k,2}}_{R_1} \right) \right) \\
&\le \mathcal{E}_{\text{off}} + \max_{R_1, R_2 \in \mathcal{C}_k(\mathcal{R})} \left(\left(V^{\widetilde{\pi}^{k,1}}_{R_1} - V^{\widetilde{\pi}^{k,1}}_{R_2} + V^{\widetilde{\pi}^{k,2}}_{R_2} - V^{\widetilde{\pi}^{k,2}}_{R_1} \right) + 4\mathcal{E}_{\text{off}} \right) \\
&= 5\mathcal{E}_{\text{off}} + \max_{R_1, R_2 \in \mathcal{C}_k(\mathcal{R})} \left(V^{\widetilde{\pi}^{k,1}}_{R_1} - V^{\widetilde{\pi}^{k,1}}_{R_2} + V^{\widetilde{\pi}^{k,2}}_{R_2} - V^{\widetilde{\pi}^{k,2}}_{R_1} \right). && (30)
\end{aligned}
$$

Consider the online preference-based regret as

$$
\text{Reg}(K) := \frac{1}{2} \sum_{k=1}^{K} \left(V^{\pi^\star} - V^{\widetilde{\pi}^{k,1}} + V^{\pi^\star} - V^{\widetilde{\pi}^{k,2}} \right), \tag{31}
$$

we have

$$
\begin{aligned}
&\text{Reg}(K) && (32)\\
&\le \sum_{k=1}^{K} \max_{R_1, R_2 \in \mathcal{C}_k(\mathcal{R})} \left(V^{\widetilde{\pi}^{k,1}}_{R_1} - V^{\widetilde{\pi}^{k,1}}_{R_2} + V^{\widetilde{\pi}^{k,2}}_{R_2} - V^{\widetilde{\pi}^{k,2}}_{R_1} \right) + 5K\mathcal{E}_{\text{off}} \\
&= \sum_{k=1}^{K} \max_{R_1, R_2 \in \mathcal{C}_k(\mathcal{R})} \Big\{ \big(V_{R_1}(\tau^{k,1}) - V_{R_2}(\tau^{k,1}) + V_{R_2}(\tau^{k,2}) - V_{R_1}(\tau^{k,2}) \big) + \\
&\quad + (V^{\widetilde{\pi}^{k,1}}_{R_1} - V_{R_1}(\tau^{k,1})) - (V^{\widetilde{\pi}^{k,1}}_{R_2} - V_{R_2}(\tau^{k,1})) \\
&\quad + (V^{\widetilde{\pi}^{k,2}}_{R_1} - V_{R_1}(\tau^{k,2})) - (V^{\widetilde{\pi}^{k,2}}_{R_2} - V_{R_2}(\tau^{k,2})) \Big\} + 5K\mathcal{E}_{\text{off}} \\
&\le \sum_{k=1}^{K} \max_{R_1, R_2 \in \mathcal{C}_k(\mathcal{R})} \big(V_{R_1}(\tau^{k,1}) - V_{R_2}(\tau^{k,1}) + V_{R_2}(\tau^{k,2}) - V_{R_1}(\tau^{k,2}) \big) \\
&\quad + 16\sqrt{\frac{K}{1-\gamma} \log(\frac{4}{\delta})} + 5K\mathcal{E}_{\text{off}} \\
&= \sum_{k=1}^{K} \max_{R_1, R_2 \in \mathcal{C}_k(\mathcal{R})} \big((R_1(\tau^{k,1}) - R_1(\tau^{k,2})) - (R_2(\tau^{k,1}) - R_2(\tau^{k,2})) \big) && (33)\\
&\quad + 16\sqrt{\frac{K}{1-\gamma} \log(\frac{4}{\delta})} + 5K\mathcal{E}_{\text{off}} \\
&\le c_1 \sqrt{\kappa d_{\Delta\mathcal{R}} K \log(K|\Delta\mathcal{R}|/\delta)} + 16\sqrt{\frac{K}{1-\gamma} \log(\frac{4}{\delta})} + 5K\mathcal{E}_{\text{off}}. && (34)
\end{aligned}
$$

The first inequality follows from Equation 30. The second inequality follows from Azuma-Hoeffding's inequality and the fact that $V_R(\tau) - V_R^{\pi}$ is a martingale when $\tau \sim \pi$. Please refer to Cai et al. (2020) for a detailed derivation. The last inequality follows directly from Lemma 10.

Finally, set $\delta = 1/K$ and follow a standard argument for regret to PAC conversion (Jin et al., 2018), we can show that the expected suboptimality of average policy $\bar{\pi}$ generated by Algorithm 2 is upper bounded by

$$
\text{SubOpt}(\bar{\pi}) \le c_0 \cdot \sqrt{\frac{C^\dagger \log(N|\mathcal{V}||\Pi|)}{N(1-\gamma)^2}} + c_1 \cdot \sqrt{\frac{d_{\text{Elu}}(\Delta\mathcal{R}, 1/K) \log(K|\Delta\mathcal{R}|)}{K(1-\gamma)}}.
$$

□

C Auxiliary Lemmas

Lemma 9. *With probability at least $1-\delta$, the following event $\mathcal{E}_1$ holds*

$$R^\star \in \mathcal{C}_k(\mathcal{R}), \quad \forall k \in [K],$$

where

$$\mathcal{C}_k(\mathcal{R}) = \left\{ R \in \mathcal{R} : ((\widehat{R}(\tau_1) - \widehat{R}(\tau_2)) - (R(\tau_1) - R(\tau_2)))^2 \le c\kappa \log(K|\Delta\mathcal{R}|/\delta) \right\},$$

c is an absolute constant and $\kappa := \frac{1}{\sigma'(2R_{max})}$ is the degree of non-linearity of the link function σ.

Proof. Using Lemma 14, we have that

$$\sum_{i=1}^{k} \left\| \mathbb{P}(\tau_k^1 \succ \tau_k^2|\widehat{R}) - \mathbb{P}(\tau_k^1 \succ \tau_k^2|R^\star) \right\|_{\text{TV}}^2 \le 2\log(|\Delta\mathcal{R}|/\delta).$$

Note that $\mathbb{P}(\tau_k^1 \succ \tau_k^2|R) = \sigma(R(\tau_1) - R(\tau_2))$, and $R(\tau)$ is bounded by $R_{\max}$, we have

$$\sum_{i=1}^{k} ((\widehat{R}(\tau_1) - \widehat{R}(\tau_2)) - (R^\star(\tau_1) - R^\star(\tau_2)))^2 \le c\kappa \log(|\Delta\mathcal{R}|/\delta)$$

Then, by the union bound, we have the conclusion immediately. □

Lemma 10. *Under event $\mathcal{E}_1$ in Lemma 9, it holds that*

$$\sum_{k=1}^{K} \left|(R_1(\tau^{k,1}) - R_1(\tau^{k,2})) - (R_2(\tau^{k,1}) - R_2(\tau^{k,2}))\right| \le O\left(\sqrt{d_{Elu}(\Delta\mathcal{R}, \delta) K \log(K|\Delta\mathcal{R}|/\delta)}\right). \tag{35}$$

Proof. Under event $\mathcal{E}_1$, we have $\max_{1\le k\le K} \operatorname{diam}(\mathcal{B}_{(\tau_1,\tau_2)_{1:k}}(\mathcal{C}_k(\mathcal{R}))) \le 2\sqrt{\kappa \log(K|\Delta\mathcal{R}|/\delta)}$ by Lemma 14, where

$$\mathcal{B}_{(\tau_1,\tau_2)_{1:k}}(\mathcal{F}) := \sup_{f_1,f_2\in\mathcal{F}} \left(\sum_{t=1}^{k} ((f_1(\tau_1^t) - f_1(\tau_2^t)) - (f_2(\tau_1^t) - f_2(\tau_2^t)))^2 \right)^{1/2}.$$

Therefore, following Lemma 11, we have

$$\begin{aligned} &\sum_{k=1}^{K} \left|(R_1(\tau^{k,1}) - R_1(\tau^{k,2})) - (R_2(\tau^{k,1}) - R_2(\tau^{k,2}))\right| \\ &\le \sum_{k=1}^{K} \mathcal{B}_{(\tau_1^k,\tau_2^k)}(\mathcal{R}_k) \\ &\le O\left(\sqrt{d_{\text{Elu}}(\Delta\mathcal{R}, \delta) K \log(K|\Delta\mathcal{R}|/\delta)}\right). \end{aligned} \tag{36}$$

□

Lemma 11 (Lemma 5 of Russo & Van Roy (2014))**.** . *Let $\mathcal{V} \in \mathcal{B}_\infty(\mathcal{X}, C)$ be a set of functions bounded by $C > 0$, $(\mathcal{V}_t)_{t\ge1}$ and $(x_t)_{t\ge1}$ be sequences such that $\mathcal{V}_t \subseteq \mathcal{V}$ and $x_t \in \mathcal{X}$ hold for $t \ge 1$. Let $\mathcal{V}|_{x_{1:t}} = \{(f(x_1), \ldots, f(x_t)) : f \in \mathcal{V}\} (\subseteq \mathbb{R}^t)$ and for $S \subseteq \mathbb{R}$, let $\operatorname{diam}(S) = \sup_{u,v\in S} \|u-v\|_2$ be the diameter of S. Then, for any $T \ge 1$ and $\alpha > 0$ it holds that*

$$\sum_{t=1}^{T} \operatorname{diam}(\mathcal{V}_t|_{x_t}) \le \alpha + C(d \wedge T) + 2\delta_T\sqrt{dT}, \tag{37}$$

where $\delta_T = \max_{1\le t\le T} \operatorname{diam}(\mathcal{V}_t|_{x_{1:t}})$ and $d = \dim_\epsilon(\mathcal{V}, \alpha)$.

The following lemmas summarizes the results regarding General Function Estimator.

Lemma 12 (Theorem A.1 in Xie et al. (2021))**.** *For any* $\pi \in \Pi$*, let* q_π *be defined as follows,*

$$q_\pi \coloneqq \arg\min_{q\in\mathcal{Q}} \sup_{\textit{admissible } \nu} \|q - T^\pi q\|_{2,\nu}^2. \tag{38}$$

Then the following event $\mathcal{E}_2$ *holds with probability as least* $1-\delta$*:*

$$\mathcal{E}(q_\pi, \pi; \mathcal{D}) \le \frac{139 \log \frac{|\mathcal{Q}||\Pi|}{\delta}}{n(1-\gamma)}, \tag{39}$$

where $\mathcal{E}(q, \pi; \mathcal{D}) \coloneqq \mathcal{L}(q, q, \pi; \mathcal{D}) - \min_{q'\in\mathcal{V}} \mathcal{L}(q', q, \pi; \mathcal{D})$.

The following lemma shows that $\mathcal{E}(q, \pi; \mathcal{D})$ could effectively estimate $\|q - T^\pi q\|_{2,\mu}^2$.

Lemma 13 (Theorem A.2 in Xie et al. (2021))**.** *For any* $\pi \in \Pi, q \in \mathcal{Q}, h \in [H]$*, and any* $\epsilon > 0$*, if* $\mathcal{E}(q, \pi; \mathcal{D}) \le \epsilon$*, Then the following event* $\mathcal{E}_3$ *holds with probability as least* $1-\delta$*:*

$$\|q - T^\pi q'\|_{2,\mu} \le \sqrt{\frac{231 \log \frac{|\mathcal{Q}||\Pi|}{\delta}}{n(1-\gamma)}} + \sqrt{\epsilon} \coloneqq \epsilon_b. \tag{40}$$

Lemma 14 (Theorem 21 in Agarwal et al. (2020))**.** *Fix* $\delta \in (0,1)$*, assume* $|\mathcal{F}| < \infty$ *and* $f^\star \in \mathcal{F}$*. Then with probability at least* $1-\delta$

$$\sum_{i=1}^n \mathbb{E}_{x\sim\mathcal{D}_i} \left\| \widehat{f}(x,\cdot) - f^\star(x,\cdot) \right\|_{TV}^2 \le 2\log(|\mathcal{F}|/\delta).$$

D ADDITIONAL EXPERIMENTAL RESULTS

Experiments on Meta-World Table 6 shows the complete experimental results in Meta-World.

Task	OPRL	PT	PT+PDS	IDRL	OPRIDE
assembly-v2	10.1±0.5	10.2±0.7	12.8±0.6	10.3±1.9	**14.2±1.3**
basketball-v2	11.7±10.2	80.7±0.1	78.7±2.0	**82.7±2.5**	61.4±2.3
bin-picking-v2	82.0±5.6	31.9±16.2	53.4±19.0	84.7±2.9	**93.3±3.2**
button-press-wall-v2	51.7±1.6	58.8±0.9	59.4±0.9	69.0±1.0	**77.7±0.1**
box-close-v2	15.0±0.7	**17.7±0.1**	17.2±0.3	16.9±0.6	16.8±0.4
coffee-push-v2	1.7±1.7	1.3±0.5	1.3±0.5	42.0±3.8	**59.4±24.8**
disassemble-v2	8.4±0.8	6.0±0.4	7.6±0.2	7.4±1.9	**12.4±2.9**
door-close-v2	61.2±1.3	65.1±10.1	62.4±8.7	78.1±3.2	**94.8±1.1**
door-unlock-v2	**79.2±2.3**	73.7±5.4	73.6±4.8	71.2±2.9	71.0±2.3
drawer-open-v2	53.0±3.3	59.7±1.3	58.3±0.1	62.5±2.0	**68.7±3.0**
faucet-close-v2	60.8±1.0	57.8±0.9	46.2±0.2	61.5±3.2	**73.1±0.8**
hammer-v2	16.4±1.0	30.2±1.7	32.6±0.8	33.6±2.8	**39.2±11.2**
hand-insert-v2	5.2±3.2	18.7±0.1	20.3±0.6	41.9±2.7	**61.8±4.9**
handle-press-v2	**28.7±4.0**	27.9±0.2	28.2±0.2	28.0±0.4	**28.7±0.1**
lever-pull-v2	**63.2±10.4**	49.2±3.7	51.7±0.1	33.1±1.2	51.8±1.6
peg-insert-side-v2	3.5±1.8	16.8±0.1	12.4±1.4	67.4±0.1	**79.0±0.2**
plate-slide-v2	77.4±1.6	4.9±0.0	37.3±2.3	**79.6±3.5**	**79.9±4.6**
push-v2	10.6±1.5	16.7±5.0	1.8±0.4	30.7±5.3	**39.3±3.4**
push-back-v2	0.8±0.0	1.1±0.4	1.1±0.1	14.0±1.1	**17.7±2.0**
push-wall-v2	7.4±4.2	74.8±14.4	3.4±0.9	89.2±3.2	**102.2±1.2**
reach-v2	63.5±2.9	82.0±0.8	84.3±0.9	75.8±1.8	**88.0±0.5**
soccer-v2	34.3±4.0	**51.3±4.1**	41.5±11.9	44.3±2.1	45.4±3.9
sweep-into-v2	37.1±13.9	9.8±0.2	9.2±0.1	63.1±3.5	**71.6±0.1**
sweep-v2	6.8±1.8	8.0±0.4	8.0±0.1	73.0±2.8	**78.5±1.0**

Table 6: Performance of offline RL algorithm on the reward-labeled dataset with different preference reward learning methods on the Meta-World tasks.

Experiments on Mujoco and Kitchen We conduct a wider range of experiments on MuJoCo and Kitchen tasks. The experimental results in Table 7 show that OPRIDE achieves superior performance compared with other baselines. The experimental results also demonstrate that the In-Dataset Exploration and Variance-based Discount Scheduling mechanisms we proposed can be effectively applied to different tasks.

Domain	Tasks	OPRL	PT	PT+PDS	OPRIDE
Mujoco	hopper-medium	23.0±0.1	36.9±2.1	35.8±1.8	**38.5±2.2**
	hopper-medium-expert	57.7±23.7	68.0±2.6	69.1±1.7	**92.3±15.8**
	walker2d-medium	70.6±1.1	**71.7±2.6**	70.9±1.8	**72.7±1.8**
	walker2d-medium-expert	108.3±3.8	**109.4±0.3**	108.4±0.5	**110.3±0.2**
	halfcheetah-medium	41.9±0.1	**42.1±0.1**	41.5±0.1	**42.4±0.1**
	halfcheetah-medium-expert	81.8±0.6	81.9±0.1	82.4±0.2	**86.5±1.5**
	kitchen-partial	34.6±0.2	48.2±4.1	**51.1±2.3**	38.7±3.7
	kitchen-mixed	46.9±0.1	42.5±1.0	44.9±1.9	**49.8±0.1**
	kitchen-partial	**62.6±1.7**	47.5±2.5	49.8±4.5	**63.7±1.1**

Table 7: Performance of offline RL algorithm on the reward-labeled dataset with different preference reward learning methods on the Mujoco tasks.

Ablation about In-Dataset Exploration module The choice to emphasize value functions over reward functions is crucial due to their ability to guide policy optimization effectively. Intuitively, while maximizing the information gain concerning the reward function (e.g., difference over the reward function) can help learn a well-calibrated reward function, it can still be sample inefficient in determining the optimal policy since we are not interested in the accuracy of the reward function in low-return regions. For instance, suppose we have actions a_1 and a_2 that lead to a terminal state s_0, and their immediate rewards are highly uncertain, ranging from $[-1, 1]$. And we have actions a_3 and a_4 that lead to high return states s_1 but yield a known fixed immediate reward of zero. By maximizing the reward differences, we will compare a_1 and a_2, but such comparison contains no information in determining the optimal policy, which will not choose a_1 and a_2 at all. Theoretically, maximizing the information gain with respect to the reward function is insufficient to derive a performance guarantee for PbRL.

We conduct additional ablation studies for these two mechanisms. The experimental results in Table 8 show that maximizing information gain about the optimal policy can achieve better performance than the reward function.

Domain	Tasks	OPRIDE (Reward Difference)	OPRIDE (Value Function Difference)
Metaworld	bin-picking	78.5±17.8	**93.3**±**3.2**
	button-press-wall	67.4±5.4	**77.7**±**0.1**
	door-close	88.3±0.7	**94.8**±**1.1**
	faucet-close	48.7±0.6	**73.1**±**0.8**
	peg-insert-side	9.7±8.5	**79.0**±**0.2**
	reach	**86.6**±**0.1**	**88.0**±**0.5**
	sweep	18.2±2.9	**78.5**±**1.0**

Table 8: Ablation study on the metaworld tasks.

Ablation about Variance-based Discount Scheduling module The choice of using a pessimistic discount factor in offline RL draws on theoretical guarantees discussed in prior works (Jiang et al., 2015; Hu et al., 2022). While prior methods may utilize a smaller fixed discount factor (Jiang et al., 2015) or tuned values in imitation learning (Liu et al., 2023), our approach innovatively employs variance-based discount scheduling to mitigate reward overestimation issues specific to offline Preference-based RL.

A smaller discount factor serves a dual purpose: it regulates optimality against sample efficiency trade-offs (Hu et al., 2022) and aligns with model-based pessimism principles, ensuring robust policy learning. Conversely, multiplicative adjustments to rewards lack theoretical grounding and often yield suboptimal performance, as evidenced in Table 9.

Domain	Tasks	OPRIDE (Penalise Reward)	OPRIDE (Penalise Discount Factor)
Metaworld	bin-picking	53.4±19.0	**93.3**±**3.2**
	button-press-wall	59.4±0.9	**77.7**±**0.1**
	door-close	62.4±8.7	**94.8**±**1.1**
	faucet-close	46.2±0.2	**73.1**±**0.8**
	peg-insert-side	12.4±1.4	**79.0**±**0.2**
	reach	84.3±0.9	**88.0**±**0.5**
	sweep	8.0±0.1	**78.5**±**1.0**

Table 9: Ablation studies about penalizing rewards and the discount factor.

E EXPERIMENT DETAILS

Experimental Setup For the Meta-World tasks, each dataset consists of 1000 trajectories. 50 trajectories of which are collected by the corresponding scripted policy added with a Gaussian noise $\mathcal{N}(0, 0.8)$ to increase diversity, and the rest 950 trajectories are collected with a policy that is a ϵ-greedy variant to the former noisy policy and select random actions with probability $\epsilon = 0.8$. For the Antmaze tasks, we use the standard dataset in the D4RL benchmark but remove the reward labels.

OPRL We use the official implementation [1], which uses 7 ensembles. Each ensemble is initially trained with 1 randomly selected query and then performs 3 rounds of active querying and training, and in each round, 1 query is acquired, making a total of 10 queries.

PT We use the official implementation [2]. We follow its original hyper-parameter settings, and change the number of queries to 10.

OPRIDE Our code is built on PT. We use the same transformer architecture and hyper-parameter with PT. The ensemble number N is 2. The size of $\mathcal{D}$ is 10000. The offline pre-training step for $V_i(\cdot, \cdot)$ in the Equation 7 is $10000 \times c$, where c is the c-th selected query. Please refer to Table 10 for detailed parameters.

Hyperparameter	Value
Optimizer	Adam
Critic learning rate	3e-4
Actor learning rate	3e-4
Mini-batch size	256
Discount factor	0.99
Target update rate	5e-3
IQL parameter τ	0.7
IQL parameter α	3.0
Query Number	10
OPRL	**Value**
Ensemble Number	7
OPRIDE	**Value**
Ensemble Number N	2
Size of $\mathcal{D}$	10000
Offline Pre-training step	$10000 \times c$
Top $m\%$	Top 30%
γ_{small}	0.7

Table 10: Hyper-parameters sheet of Algorithms.

[1] https://github.com/danielshin1/oprl
[2] https://github.com/csmile-1006/PreferenceTransformer/tree/main