THE PROVABLE BENEFITS OF UNSUPERVISED DATA SHARING FOR OFFLINE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-supervised methods play a vital role in fueling the progress of deep learning using supervision from the data itself, reducing the need for expensive annotations. Like self-supervised methods in other areas of machine learning, we would like methods that can leverage reward-free data to improve the performance of offline RL, but it is unclear how to conduct such self-supervised RL in a principled way. In this paper, we examine the theoretical benefit of unlabeled data in the context of linear MDPs and propose a novel and **P**rovable **D**ata **S**haring algorithm, which we refer to as PDS, to utilize such unlabeled data for offline RL. PDS utilizes additional penalties upon the reward function learned from labeled data to avoid potential overestimation of the reward. We show that such a penalty is crucial to keep the algorithm conservative, and PDS achieves a provable benefit from unlabeled data under mild conditions. We conduct extensive experiments on various offline RL tasks and show that PDS can significantly improve offline RL algorithms with unlabeled data.

1 INTRODUCTION

Offline reinforcement learning (RL) offers a promising framework to learn performant sequential policies with pre-collected datasets, which is highly preferred to online RL in many real-world problems where active data collection and exploration is expensive, unsafe, or infeasible (Swaminathan & Joachims, 2015; Shalev-Shwartz et al., 2016; Singh et al., 2020). However, this approach can still be expensive since labeling large datasets with rewards can be costly and requires huge human efforts (Singh et al., 2019; Wirth et al., 2017). On the contrary, unlabeled data can be cheap and abundant, encouraging us to learn from self-supervisions with unlabeled data. Such self-supervised methods have achieved great success in computer vision and natural language processing tasks like Brown et al. (2020); Devlin et al. (2018); Chen et al. (2020), but it is less discussed in the context of offline reinforcement learning.

Some prior works (Ho & Ermon, 2016; Reddy et al., 2019; Kostrikov et al., 2019) have explored demonstration-based approaches, which remove the need for reward annotations but require the samples to be near optimal. There is also a line of work that focuses on data sharing from different datasets (Kalashnikov et al., 2021; Yu et al., 2021a) but assumes that the dataset can be relabeled with oracle reward functions for the target task. These assumptions can be strong in real-world problems, where both expert trajectories and reward labeling are expensive. It is important but non-trivial to incorporate unlabeled datasets into offline RL due to RL problems' sequential and dynamic nature. Prior work (Yu et al., 2022) has shown that learning to predict the rewards can be challenging and that the simple approach of setting the reward to zero can achieve good results. However, it is unclear what affects the performance of reward prediction methods and whether reward-free data can be provably beneficial to offline RL, which naturally leads to the following question:

How can we leverage reward-free data to improve the performance of offline RL algorithms in a principled way?

To answer this question, we start with the analysis of the theoretical benefit of using unlabeled data in linear MDPs. Intuitively, unlabeled data helps us gain more information about the dynamics of the MDP, leaving only uncertainty over reward functions. ~~As a result, the problem is reduced to a linear bandit problem with uncertainty only on reward functions when the unlabeled data goes~~

~~to infinity.~~ Based on this analysis, we may develop a model-based approach and add pessimism over models and rewards, respectively, but it can be computationally extensive. To sidestep this issue, we propose a model-free method, which we refer to as provable data sharing (PDS), which simply adds uncertainty penalties on learned rewards to keep the algorithm conservative. We show that our method can cooperate with model-free offline methods while being simple and provably efficient. Theoretical analysis shows that our method can improve performance bounds under mild conditions. In practice, our method also achieves superior performance. We conduct extensive experiments on various environments, including single-task domains like MuJoCo (Todorov et al., 2012) and kitchen, as well as multi-task domains like AntMaze and Meta-World (Yu et al., 2020a). Results show that our algorithm improves significantly over previous methods like UDS (Yu et al., 2022) and naive reward prediction methods.

Our main contribution is the provable data sharing (PDS) algorithm, a novel method to utilize unsupervised data for offline RL with theoretical guarantees. PDS is a model-free method that adds uncertainty penalties upon learned reward functions and can be easily integrated with existing offline RL algorithms. We show that PDS can achieve provable improvement in linear MDPs under mild conditions. We also show that our method achieves superior performance in practice with extensive experiments on various tasks like locomotion, navigation, and manipulation tasks.

2 Related Work

Offline Reinforcement Learning Current offline RL methods (Levine et al., 2020) can be roughly divided into policy constraint-based, uncertainty estimation-based, and model-based approaches. Policy constraint methods aim to keep the policy close to the behavior under a probabilistic distance (Siegel et al., 2020; Yang et al., 2021; Kumar et al., 2020; Fujimoto & Gu, 2021; Fujimoto et al., 2019; Nair et al., 2020; Hu et al., 2022; Peng et al., 2019; Kostrikov et al., 2021; Ma et al., 2021). Uncertainty estimation-based methods attempt to consider the Q-value prediction's confidence using dropout or ensemble techniques (An et al., 2021; Wu et al., 2021). Last, model-based methods incorporates the uncertainty in the model space for conservative offline learning (Yu et al., 2020b; 2021b; Kidambi et al., 2020).

Offline Data Sharing Prior works have demonstrated that data sharing across tasks can be beneficial by designing sophisticated data-sharing protocols (Yu et al., 2021a). For example, previous studies have explored developing data sharing strategies by human effort (Kalashnikov et al., 2021), inverse RL (Reddy et al., 2019; Li et al., 2020), and estimated Q-values (Yu et al., 2021a). However, these data sharing works must assume the dataset can be relabeled with oracle rewards for the target task, which is a strong assumption since the high cost of reward labeling. Therefore, effectively incorporating the unlabeled data into the offline RL algorithms is essential. To solve this issue, some recent work (Yu et al., 2022) proposes simply applying zero rewards to unlabeled data. In this work, we propose a principle way to leverage unlabeled data without the strong assumption of reward relabeling.

Reward Prediction It is widely observed that reward shaping and intrinsic rewards can accelerate learning in online RL (Mataric, 1994; Ng et al., 1999; Wu & Tian, 2017; Song et al., 2019; Guo et al., 2016; Abel et al., 2021). There are also extensive works that studies automatically designing reward functions using inverse RL (Ng et al., 2000; Fu et al., 2017). However, there is less attention on the offline setting where online interaction is not allowed and the trajectories may not be optimal.

3 Preliminaries

3.1 Linear MDPs and Performance Metric

We consider infinite-horizon discounted Markov Decision Processes (MDPs), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$, with state space $\mathcal{S}$, action space $\mathcal{A}$, discount factor $\gamma \in [0, 1)$, transition function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, and reward function $r : \mathcal{S} \times \mathcal{A} \to [0, r_{\max}]$. To make things more concrete, we consider the *linear MDP* (Yang & Wang, 2019; Jin et al., 2020) as follows, where the transition kernel and expected reward function are linear with respect to a feature map.

Definition 3.1 (Linear MDP)**.** We say an episodic MDP $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$ is a linear MDP with known feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ if there exist unknown measures $\mu = (\mu_1, \ldots, \mu_d)$ over $\mathcal{S}$ and an unknown vector $\theta \in \mathbb{R}^d$ such that

$$\mathcal{P}(s' \mid s, a) = \langle \phi(s, a), \mu(s') \rangle, \quad r(s, a) = \langle \phi(s, a), \theta \rangle \tag{1}$$

for all $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$. And we assume $\|\phi(s, a)\|_2 \leq 1$ for all $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ and $\max\{\|\mu(\mathcal{S})\|_2, \|\theta\|_2\} \leq \sqrt{d}$, where $\|\mu(\mathcal{S})\| \equiv \int_{\mathcal{S}} \|\mu(s)\| ds$.

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, i.e., $a_t \sim \pi(\cdot \mid s_t)$. 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

$$V^\pi(s) = \mathbb{E}_\pi \Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \,\Big|\, s_0 = s \Big], \quad Q^\pi(s, a) = \mathbb{E}_\pi \Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \,\Big|\, s_0 = s, a_0 = a \Big]. \tag{2}$$

where the expectation is with respect to the trajectory τ induced by policy π.

We define the Bellman operator as

$$(\mathbb{B} f)(s, a) = \mathbb{E}_{s' \sim p(\cdot \mid s, a)} \big[r(s, a) + \gamma f(s') \big]. \tag{3}$$

We use π^*, Q^*, and V^* to denote the optimal policy, optimal Q-function, and optimal value function, respectively. We have the Bellman optimality equation

$$V^*(s) = \max_{a \in \mathcal{A}} Q^*(s, a), \quad Q^*(s, a) = (\mathbb{B} V^*)(s, a). \tag{4}$$

Meanwhile, the optimal policy π^* satisfies

$$\pi^*(\cdot \mid s) = \operatorname*{argmax}_{\pi} \langle Q^*(s, \cdot), \pi(\cdot \mid s) \rangle_{\mathcal{A}}, \quad V^*(s) = \langle Q^*(s, \cdot), \pi^*(\cdot \mid s) \rangle_{\mathcal{A}},$$

where the maximum is taken over all functions mapping from $\mathcal{S}$ to distributions over $\mathcal{A}$. We aim to learn a policy that maximizes the expected cumulative reward. Correspondingly, we define the performance metric as

$$\text{SubOpt}(\pi, s) = V^{\pi^*}(s) - V^\pi(s). \tag{5}$$

3.2 Provable Offline Algorithms

In this section, we consider *pessimistic value iteration* (PEVI; Jin et al., 2021) as the backbone algorithm, and we provide a detailed description in Algorithm 3 in Appendix A. 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 $\mathcal{D}$. Please refer to Appendix A for details of the PEVI algorithm.

We use the notion of ξ-uncertainty quantifier as follows to formalize the idea of pessimism.

Definition 3.2 (ξ-Uncertainty Quantifier)**.** We say $\Gamma : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a ξ-uncertainty quantifier for $\widehat{\mathbb{B}}$ and $\widehat{V}$ if with probability $1 - \xi$, for all $(s, a) \in \mathcal{S} \times \mathcal{A}$,

$$\big| (\widehat{\mathbb{B}} \widehat{V})(s, a) - (\mathbb{B} \widehat{V})(s, a) \big| \leq \Gamma(s, a). \tag{6}$$

3.3 Unsupervised Data Sharing

We consider unsupervised data sharing in offline reinforcement learning. We first characterize the quality of the dataset with the notion of coverage coefficient (Uehara & Sun, 2021), defined as below.

Definition 3.3. The coverage coefficient $C^\dagger$ of a dataset $\mathcal{D} = \{(s_\tau, a_\tau, r_\tau)\}_{\tau=1}^N$ is defined as

$$C^\dagger = \sup_C \left\{ \frac{1}{N} \cdot \sum_{\tau=1}^{N} \phi(s_\tau, a_\tau) \phi(s_\tau, a_\tau)^\top \succeq C \cdot \mathbb{E}_{\pi^*} \big[\phi(s_t, a_t) \phi(s_t, a_t)^\top \,\big|\, s_0 = s \big], \forall s \in \mathcal{S} \right\},$$

The coverage coefficient $C^\dagger$ represents the maximum ratio between the density of empirical state-action distribution and the density induced from the optimal policy. Intuitively, it represents the quality of the dataset. For example, the `expert` dataset has a high coverage ratio while the `random` dataset may have a low ratio.

We denote $\mathcal{D}_0$ as the origin labeled dataset, with coverage coefficient $C_0^\dagger$ and size N_0. And we denote the unlabeled dataset as $\mathcal{D}_1$, with coverage coefficient $C_1^\dagger$ and size N_1. Note that it is possible that the unlabeled data comes from multiple sources, such as multi-task settings, and we still use $\mathcal{D}_1$ to represent the combined dataset $\cup_{i=1}^M \mathcal{D}_i$ for simplicity.

4 Provable Unsupervised Data Sharing

How can we leverage unlabeled data for offline RL? At first sight, we may learn the reward function from labeled data via the following regression

$$\widehat{\theta} = \operatorname*{argmin}_{\theta} \sum_{\tau=1}^{N_0} \left(f_\theta(s_\tau, a_\tau) - r_\tau\right)^2 + \frac{\nu}{2}\|\theta\|_2^2, \tag{7}$$

where $f_\theta(s_\tau, a_\tau) = \phi(s_\tau, a_\tau)^\top \theta$ in linear MDPs. Then we can use this learned reward function to label unsupervised data. However, Yu et al. (2022) observes that using such a naive reward prediction method can lead to suboptimal performances, which is even worse than using all zeros in some scenarios. This is due to the potential overestimation of the predicted reward $r_{\widehat{\theta}}$, which ruins the pessimism in offline algorithms.

To address this issue, we first analyze the uncertainty in learned reward functions. We propose simply adding penalties for such uncertainty to leverage unlabeled data, which leads to a provable data-sharing algorithm, which we refer to as PDS. As we will show in Section 4.2, PDS has a provable performance bound consisting of two parts. The first part is the offline error, which is tightened compared to no data sharing because of additional data, and the second part comes from the uncertainties in rewards. As long as the unlabeled dataset has mediocre size or quality, the performance bound is provably better than no data sharing. We extend our algorithm in linear MDPs to general settings in Section 4.3 and propose to use ensembles for the reward uncertainty estimation. We use IQL (Kostrikov et al., 2021) as the backbone offline algorithm, which leads to Algorithm 2. The algorithm is simple and can be easily integrated with other model-free offline algorithms.

4.1 Provable Data Sharing

We begin by analyzing the uncertainty in learned reward functions. In the context of linear MDPs, the reward function can be learned via linear regression, and the uncertainty of the parameters is characterized by the following lemma.

Lemma 4.1 (Abbasi-Yadkori et al. (2011)). *Let* $\alpha = \sqrt{\nu} + r_{\max} \cdot \sqrt{2\log\frac{1}{\delta} + d\log\left(1 + \frac{N_0}{\nu d}\right)}$, $\Lambda = \nu I + \sum_{\tau=1}^{N_0} \phi(s_\tau, a_\tau)\phi(s_\tau, a_\tau)^\top$,

$$\mathcal{C}(\delta) = \left\{\theta \in \mathbb{R}^d \mid \|\theta - \widehat{\theta}\|_\Lambda \le \alpha\right\}, \tag{8}$$

where $\widehat{\theta}$ is the minimizer in Equation (7), then we have $\mathcal{P}(\theta^\star \in \mathcal{C}(\delta)) \ge 1 - \delta$, where $\theta^\star$ is the true parameter for the reward function.

Proof. Please refer to Theorem 2 in Abbasi-Yadkori et al. (2011) for detailed proof. □

Lemma 4.1 is a standard ellipsoid argument and shows that with high probability, the true parameters lie in the range of an ellipsoid near the learned parameters. The diameter of the ellipsoid only depends on the quality and size of labeled data. This motivates us to give a more accurate estimation of the reward function to keep the overall algorithm pessimistic while keeping the reward bias small. The simple reward prediction method ruins the pessimistic estimation of the algorithm and UDS(Yu et al., 2022) may induce a too-large reward bias. As we will show in Theorem 4.3, we can add

pessimism over the reward function and plug in other model-free algorithms like PEVI (Jin et al., 2021), and we will have a provable performance bound. Based on this, we design an algorithm consisting of two phases. In the first phase, we construct a pessimistic reward estimator by finding the reward function in the confidence set that leads to the lowest optimal value. In the second phase, our algorithm conducts standard offline RL with the given pessimistic reward function. The overall algorithm is summarized in Algorithm 1.

It is still computationally challenging to find the best parameter in the confidence set as in Equation 11, which is a bi-level optimization problem since it requires computing a value estimation for each θ and minimizing over θ. To solve this challenge, we propose simply using a pessimistic estimation, which keeps the pessimistic property of the offline algorithms. Formally,

$$\widehat{r}(s,a) = \max\left\{\phi(s,a)^\top \widehat{\theta} - \alpha\sqrt{\phi(s,a)^\top \Lambda^{-1}\phi(s,a)}, 0\right\}, \tag{9}$$

where $\Lambda = \nu I + \sum_{\tau=1}^{N_0} \phi(s_\tau, a_\tau)\phi(s_\tau, a_\tau)^\top$.

We use such a pessimistic estimation because Equation (9) is a lower bound of reward functions in the confidence set $\mathcal{C}(\delta)$, which is guaranteed by the following lemma:

Lemma 4.2. *For any $\theta \in \mathcal{C}(\delta)$,*

$$\left|\phi(s,a)^\top \theta - \phi(s,a)^\top \widehat{\theta}\right| \le \alpha\sqrt{\phi(s,a)^\top \Lambda^{-1}\phi(s,a)}. \tag{10}$$

This lemma follows directly from Cauchy-Schwartz inequalities. Then it is easy to see that with a high probability of $1-\delta$, Equation (9) does not overestimate the true reward function. When the amount of labeled data is limited, or the distributional shift between labeled and unlabeled dataset is large, Equation (9) will degrade to 0, which is exactly the UDS (Yu et al., 2022) algorithm.

Algorithm 1 Provable Data Sharing, Linear MDP

1: **Require**: Labeled dataset $\mathcal{D}_0 = \{(s_\tau, a_\tau, r_\tau)\}_{\tau=1}^{N_0}$, unlabeled dataset $\mathcal{D}_1 = \{(s_\tau, a_\tau)\}_{\tau=1}^{N_1}$.
2: **Require**: Confidence parameter α, β, δ.
3: Learn the reward function $\widehat{\theta}$ from $\mathcal{D}_0$ using Equation (7)
4: Construct the confidence set $\mathcal{C}(\delta)$ using Equation (8).
5: Construct the pessimistic reward over the confidence set

$$\widetilde{\theta} \leftarrow \operatorname*{argmin}_{\theta \in \mathcal{C}(\delta)} \widehat{V}_\theta, \tag{11}$$

where $\widehat{V}_\theta$ is the estimated value function from Algorithm 3 and dataset $\mathcal{D}_0 \cup \mathcal{D}_1$ with reward relabeled with parameter θ.

6: Annotate the reward in $\mathcal{D}_0 \cup \mathcal{D}_1$ with parameter $\widetilde{\theta}$.
7: Learn the policy from the annotated dataset $\mathcal{D}_0 \cup \mathcal{D}_1$ using Algorithm 3

$$\widehat{V}, \widehat{\pi} \leftarrow \text{PEVI}(\mathcal{D}_0 \cup \mathcal{D}_1). \tag{12}$$

8: **Return** $\widehat{\pi}$

4.2 Theoretical Analysis

In this subsection, we analyze how PDS improves the performance bound with unlabeled data.

Theorem 4.3 (Provable Benefit of Unlabeled Data)**.** *Suppose the dataset $\mathcal{D}_0, \mathcal{D}_1$ have positive coverage coefficients $C_0^\dagger, C_1^\dagger$, and the underlying MDP is a linear MDP. In Algorithm 1, we set*

$$\lambda = 1, \nu = 1, \alpha = 2\sqrt{d\zeta_2}\cdot r_{\max}, \beta = \frac{cd\sqrt{\zeta_1}}{1-\gamma}\cdot r_{\max}, \zeta_1 = \log\left(\frac{4d(N_0+N_1)}{(1-\gamma)\delta}\right), \zeta_2 = \log\left(\frac{2dN_0}{\delta}\right),$$

where $c > 0$ is an absolute constant and $\delta \in (0, 1)$ is the confidence parameter. Then with probability $1 - 2\delta$, the policy $\widehat{\pi}$ generated by PDS satisfies for all $s \in \mathcal{S}$,

$$\text{SubOpt}(\widehat{\pi}; s) \leq \frac{2cr_{\max}}{(1-\gamma)^2}\sqrt{\frac{d^3\zeta_1}{N_0C_0^\dagger + N_1C_1^\dagger}} + \frac{4r_{\max}}{1-\gamma}\sqrt{\frac{d^2\zeta_2}{N_0C_0^\dagger}}.$$

Proof. Please refer to Appendix B for detailed proof. □

The performance bound of PDS consists of two terms. The first is the offline error, which inherits from offline algorithms. Compared to the bound without data sharing, this bound is improved due to additional unlabeled data with size N_1 and coverage $C_1^\dagger$. The second term is reward error, which comes from uncertainties in the rewards. Note that the second term is the same as the performance bound of a linear bandit with rewards in the range $[0, r_{\max}/(1-\gamma)]$. This means that when the number of unlabeled data goes to infinity, the uncertainty of the dynamics becomes zero, and the RL problem degenerates to linear bandits. This theorem reflects the performance bound in the worst case and improving the bound means improving the worst-case performance.

To better see the benefit of unlabeled data, we define the **suboptimality bound ratio (SBR)** of an offline algorithm $\mathcal{A}$ as the ratio of the suboptimality bound from the policy learned with additional unlabeled data to the suboptimality bound of the policy learned with only labeled data:

$$\text{SBR}(\mathcal{A}) = \frac{\overline{\text{SubOpt}}\big(\widehat{\pi}_{\mathcal{A}}(\mathcal{D}_0, \mathcal{D}_1)\big)}{\overline{\text{SubOpt}}\big(\widehat{\pi}_{\mathcal{A}}(\mathcal{D}_0, \varnothing)\big)}, \tag{13}$$

where $\overline{\text{SubOpt}}$ is an upper bound for the suboptimality. Note that the ratio compares the bound of suboptimality rather than the suboptimality itself. Intuitively, SBR measures the benefit of unlabeled data to the offline algorithm. The smaller the SBR, the more benefit the unlabeled data brings to the offline algorithm. Applying the definition above to PDS, we have the following corollary.

Corollary 4.4 (Informal)**.** *The SBR of PDS satisfies*

$$\text{SBR} \approx \underbrace{\sqrt{\frac{N_0C_0^\dagger}{N_0C_0^\dagger + N_1C_1^\dagger}}}_{\text{finite sample term}} + \underbrace{\frac{2(1-\gamma)}{c\sqrt{d}}}_{\text{asymptotic term}}, \tag{14}$$

where c is the constant in Theorem 4.3 and we ignore the logarithmic factors.

When does unlabeled data improve the performance of offline algorithms? Based on Corollary 4.4, we can analyze the relative performance of PDS under different conditions. The first term depends on the qualities and amounts of both labeled and unlabeled datasets. It will be sufficiently small if the unlabeled dataset has a mediocre number of samples or data quality. The second term affects the asymptotic performance when the unlabeled data goes to infinity. It depends on the discount factor and the dimension of the problem. The longer the horizon or, the larger the problem dimension, the better PDS improves over no data-sharing algorithms asymptotically. Please refer to Appendix F for a more detailed discussion.

4.3 Practical Implementation

This subsection discusses the practical implementation of PDS in general MDPs. For the uncertainty estimation, we choose to use L ensembles $\{\theta_1, \ldots, \theta_L\}$, which is learned via Equation (9). Then we use the following pessimistic estimation:

$$\widehat{r}(s, a) = \max\{\mu(s, a) - k\sigma(s, a), 0\}, \tag{15}$$

where $\mu(s, a) = \frac{1}{L}\sum_{i=1}^{L} f_{\theta_i}(s, a), \sigma(s, a) = \sqrt{\frac{1}{L}\sum_{i=1}^{L}(f_{\theta_i}(s, a) - \mu(s, a))^2}$ are the mean and standard deviation, respectively. k is a hyperparameter to control the amount of pessimism. Note that we can also use the minimum over L ensembles for the pessimistic estimation, which is linked to Equation (15) following An et al. (2021); Royston (1982) as

$$\mathbb{E}\left[\min_{j=1,\ldots,L} f_{\theta_j}(s, a)\right] \approx \mu(s, a) - \Phi^{-1}\left(\frac{L - \frac{\pi}{8}}{L - \frac{\pi}{4} + 1}\right)\sigma(s, a), \tag{16}$$

where Φ is the CDF of the standard Gaussian distribution. There are more advanced estimation methods like the dropout-based method (Wu et al., 2021) and the model-based method (Yu et al., 2020b), but we stick to the ensemble method for simplicity. Then we can plug in any model-free offline algorithms. Here we use IQL (Kostrikov et al., 2021) as the backbone offline algorithm for illustration, but we emphasize that it can be easily integrated with other model-free algorithms. The overall algorithm is summarized in Algorithm 2.

Algorithm 2 IQL+PDS algorithm, General MDPs

Input: Labeled dataset $\mathcal{D}_0$, unlabeled dataset $\mathcal{D}_1$.
Input: Parameter α, β, k, τ.
Output: policy π_ϕ.

Learn L reward functions as in Equation (7).
Construct pessimistic reward estimation as in Equation (15) or (16).
Relabel unsupervised dataset $\mathcal{D}_1$ and combine with the labeled dataset $\mathcal{D}_0$.
Initialize ψ, θ, $\widehat{\theta}$, ϕ.
for each gradient step **do**
  $\phi \leftarrow \psi - \lambda_V \nabla_\psi L_V(\psi), \quad L_V(\psi) = \mathbb{E}_{s,a}\left[L_2^\tau(Q_{\widehat{\theta}}(s,a) - V_\psi(s))\right]$
  $\theta \leftarrow \theta - \lambda_Q \nabla_\theta L_Q(\theta), \quad L_Q(\theta) = \mathbb{E}_{s,a,r}\left[(r + \gamma Q_{\widehat{\theta}}(s,a) - Q_\theta(s,a))^2\right]$
  $\widehat{\theta} \leftarrow \alpha\theta + (1-\alpha)\theta$
end for
for each gradient step **do**
  $\phi \leftarrow \phi - \lambda_\pi \nabla_\phi L_\pi(\phi), \quad L_\pi(\phi) = \mathbb{E}_{s,a}\left[\exp\left(\beta(Q_{\widehat{\theta}}(s,a) - V_\psi(s))\right)\log(\pi_\phi(a|s))\right]$
end for

5 EXPERIMENTS

In this section, we aim to evaluate the effectiveness of pessimistic reward estimation and answer the following questions: (1) How does PDS perform compared to the naive reward prediction and unlabeled data sharing (UDS) methods in single locomotion and manipulation tasks? (2) How does PDS behave in multi-task offline RL settings? (3) Why and when is PDS effective?

Algorithm	door-open	door-close	drawer-open	drawer-close	average
UDS	16.2±12.1	0.0±0.0	30.4±60.4	**182.2±0.4**	57.2±30.8
Prediction	**26.4±12.9**	110.8±14.3	102.6±40.2	**182.2±0.4**	105.5±22.3
PDS	**30.8±11.4**	**118.6± 2.3**	**117.8±44.5**	**182±0.0**	**112.3±23.0**
No Share	4.8±9.5	0.0±0.0	29.6±58.7	175.0±17.6	52.4±31.0
Oracle	20.6±13.3	113.2±5.7	135.6±36.5	182.6±0.4	113.0±19.6

Table 1: Experiment results for multi-task robotic manipulation (Meta-World) experiments. Numbers are averaged across five seeds and we bold the best-performing method that does not have access to the true rewards.

Env	Tasks / Dataset type	UDS	Prediction	CDS+UDS	PDS
Antmaze	medium-play (3 tasks) / directed	15.8±1.2	26.2±3.7	**40.6±4.0**	33.0±3.9
	medium-play (3 tasks) / undirected	19.6±2.5	27.2±2.9	37.2±5.1	**40.0±2.8**
	medium-diverse (3 tasks) / directed	8.7±3.3	20.2±3.8	**33.3±11.5**	**33.8±2.4**
	medium-diverse (3 tasks) / undirected	9.7±1.3	42.2±4.3	40.7±3.5	**54.5±3.6**

Table 2: Experiment results for AntMaze tasks with normalized score metric averaged with five random seeds.

Single-task domains and datasets. To answer Question (1), we perform empirical evaluations on the hopper, walker2d, and kitchen tasks adopted from D4RL (Fu et al., 2020). We consider 50

Task	Labeled Size	Unlabeled Size	UDS	Prediction	PDS	Oracle
Hopper	medium / 50K	medium / 0.1M	58.7±1.5	**64.8±3.2**	**63.5±1.9**	66.3±2.1
	medium / 50K	medium / 0.4M	57.3±1.4	**68.9±4.0**	**68.7±1.8**	67.4±4.2
	medium / 50K	medium / 0.6M	56.6±1.4	**68.5±2.1**	**68.8±0.7**	68.2±2.1
	expert / 50K	random / 0.1M	53.3±3.8	27.9±15.1	**61.2±5.9**	18.1±5.9
	random / 50K	expert / 0.1M	4.3±0.4	84.7±10.8	**93.5±4.4**	40.1±4.5
Walker2d	medium / 50K	medium / 0.1M	70.8±1.2	71.4±2.9	**74.2±1.4**	74.6±2.3
	medium / 50K	medium / 0.4M	75.3±1.4	70.9±4.1	**80.6±1.5**	77.7±2.4
	medium / 50K	medium / 0.6M	74.8±0.4	**79.9±4.2**	**81.4±0.6**	79.1±2.7
	expert / 50K	random / 0.1M	25.7±13.1	2.7±0.2	**76.4±6.9**	22.6±1.2
	random / 50K	expert / 0.1M	0.4±0.1	95.3±2.3	**104.1±3.0**	15.8±1.3

Table 3: Experimental results with normalized score metric averaged with five random seeds.

labeled trajectories and various amounts of unlabeled trajectories of different sizes and qualities. This setup resembles practical problems where the labeled data is scarce.

Multi-task domains and datasets. As for Question (2), we consider several multi-task domains. (1) The first set of domains is the Meta-World (Yu et al., 2020a). We consider the same setup as in CDS (Yu et al., 2021a) and four tasks: `door open`, `door close`, `drawer open` and `drawer close`. Each task has a sparse reward of 1 when the success condition is met and 0 otherwise. (2) The second domain is the AntMaze task in D4RL, which consists of mazes of two sizes (medium and large) with 3 and 7 tasks corresponding to different goal positions, respectively. Following Yu et al. (2021a; 2022), we construct two set of tasks, `undirected` and `directed`. The trajectories in the `undirected` dataset is randomly and uniformly divided into different tasks, while the `directed` dataset is associated with the goal closest to the final state of the trajectory. Please refer to Appendix E for detailed description.

Comparisons. We adopt the state-of-the-art unlabeled data sharing method (UDS) (Yu et al., 2022) as our baseline, and we combine UDS with IQL (Kostrikov et al., 2021), the same underlying offline RL method as PDS for a fair comparison. We also train naive reward prediction method (Prediction), sharing all true rewards method (Oracle), and adapt them with IQL. For more details on experimental settings and hyper-parameter settings, please refer to Appendix G.

5.1 Experimental Results

Results of Question (1). We evaluate each method on the hopper, walker2d, and kitchen domains. As shown in Table 3, PDS outperforms other methods on most tasks and achieves competitive or better performance than the oracle method. Particularly, PDS performs well when the labeled and unlabeled datasets have different data qualities. We hypothesize that this is because PDS successfully captures the uncertainties induced by this distribution shift and keeps the algorithm pessimistic. The prediction method performs well when the unlabeled dataset has a good quality, and UDS performs well when the unlabeled data has low quality. PDS takes the better of both worlds and achieves a better performance.

Results of Question (2). Multi-task settings exhibit more significant distributional shifts between labeled data and unlabeled data due to the different goals of the tasks. To evaluate the performance of PDS on multi-task settings, we evaluate each method on the AntMaze and the Meta-World domain, and the result is shown in Table 1 and Table 2. On Meta-World experiments, the performance of PDS is comparable to the oracle method and outperforms other methods. UDS performs relatively poorly on this dataset. We hypothesize that this is because the dataset quality is relatively high, and labeling the dataset with zeros induces a large reward bias. On the multi-task AntMaze domain, PDS outperforms both UDS and the naive prediction method. The performance gap between PDS and other baselines is larger on the diverse dataset. This aligns with our observation on single-task domains that PDS performs better when the distribution shift between datasets is larger.

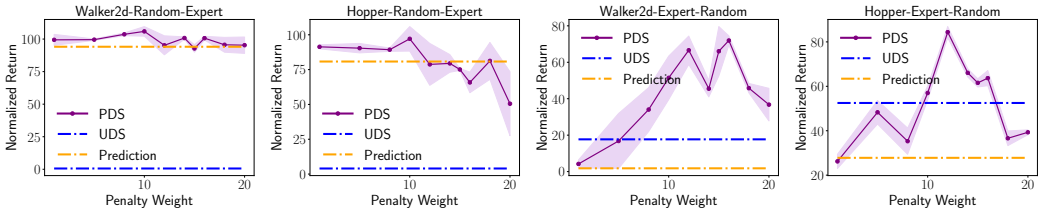


Figure 1: Impact of penalty weight k on the performance. We evaluate PDS on Hopper/Walker2d-Labeled (Expert/Random)-Unlabeled (Expert/Random) tasks with various k.

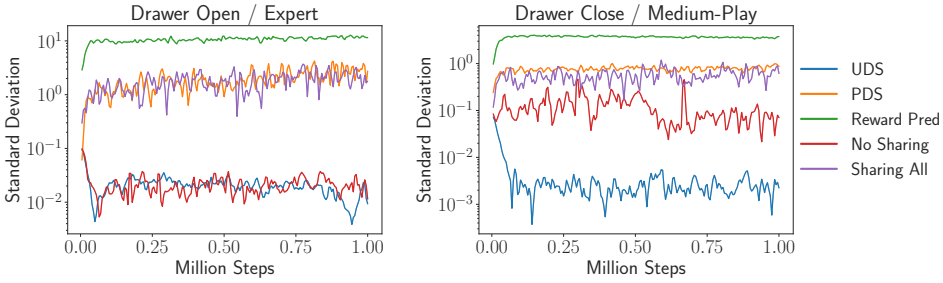


Figure 2: The standard deviation of the state values learned with IQL. The reward prediction method suffers from a large deviation. The oracle method's and PDS's deviation is mediocre, and no sharing and UDS methods enjoy the smallest deviation.

Results of Question (3). To investigate the effect of uncertainty weights, we conduct experiments on the hopper and walker2d tasks with various penalty weight k, shown in Figure 1. The results show that PDS interpolates between UDS and the reward prediction method. PDS degenerates to naive reward prediction if k is small and performs like UDS when k is relatively large. PDS offers a better trade-off to balance the conservation and generalization of reward estimators and thus achieves better performance. We plot the standard deviation of state values learned by IQL between different runs to investigate further why PDS improves over naive prediction methods. The result in Figure 2 shows that PDS reduces the variance from reward prediction and is close to the oracle method. This indicates that PDS successfully reduces the uncertainties from the variance of reward predictors while keeping the reward bias small.

Discussion between PDS, UDS and Prediction: PDS is the generalization of UDS and Prediction methods since UDS is PDS with $k = \infty$, and Prediction is PDS with $k = 0$. Further, UDS induces a high bias, and Prediction ruins the pessimism of offline algorithms. On the contrary, PDS achieves a trade-off between bias and pessimism by adaptively adjusting k. Therefore, it is not surprising that tuning k yielding better results than either baseline.

6 Conclusion

This paper investigates the theoretical benefit of using unlabeled data in offline reinforcement learning. We show that unlabeled data helps us gain more information about the dynamics of the MDP, reduce the problem to linear bandits, and improve the performance bound accordingly. Based on this analysis, we propose a new method, PDS, which adds uncertainty penalties on learned rewards to keep the algorithm conservative. We show that our method had provable guarantees in theory and achieves superior performance in practice. Integrating representation learning methods for offline unsupervised data sharing is an interesting future direction. It is also an interesting direction to extend our analysis into more general settings than linear MDPs like generalized linear MDPs (Wang et al., 2019) and low-rank MDPs (Ayoub et al., 2020; Jiang et al., 2017).

REFERENCES

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. Advances in neural information processing systems, 24:2312–2320, 2011.

David Abel, Will Dabney, Anna Harutyunyan, Mark K Ho, Michael Littman, Doina Precup, and Satinder Singh. On the expressivity of markov reward. Advances in Neural Information Processing Systems, 34:7799–7812, 2021.

Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. Advances in Neural Information Processing Systems, 34, 2021.

Alex Ayoub, Zeyu Jia, Csaba Szepesvari, Mengdi Wang, and Lin Yang. Model-based reinforcement learning with value-targeted regression. In International Conference on Machine Learning, pp. 463–474. PMLR, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In International conference on machine learning, pp. 1597–1607. PMLR, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. arXiv preprint arXiv:1710.11248, 2017.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. arXiv preprint arXiv:2004.07219, 2020.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. arXiv preprint arXiv:2106.06860, 2021.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In International Conference on Machine Learning, pp. 2052–2062. PMLR, 2019.

Xiaoxiao Guo, Satinder Singh, Richard Lewis, and Honglak Lee. Deep learning for reward design to improve monte carlo tree search in atari games. arXiv preprint arXiv:1604.07095, 2016.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. Advances in neural information processing systems, 29, 2016.

Hao Hu, Yiqin Yang, Qianchuan Zhao, and Chongjie Zhang. On the role of discount factor in offline reinforcement learning. arXiv preprint arXiv:2206.03383, 2022.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low bellman rank are pac-learnable. In International Conference on Machine Learning, pp. 1704–1713. PMLR, 2017.

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

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

Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. arXiv preprint arXiv:2104.08212, 2021.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. arXiv preprint arXiv:2005.05951, 2020.

Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. Imitation learning via off-policy distribution matching. arXiv preprint arXiv:1912.05032, 2019.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. arXiv preprint arXiv:2110.06169, 2021.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. arXiv preprint arXiv:2006.04779, 2020.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643, 2020.

Alexander Li, Lerrel Pinto, and Pieter Abbeel. Generalized hindsight for reinforcement learning. Advances in neural information processing systems, 33:7754–7767, 2020.

Xiaoteng Ma, Yiqin Yang, Hao Hu, Qihan Liu, Jun Yang, Chongjie Zhang, Qianchuan Zhao, and Bin Liang. Offline reinforcement learning with value-based episodic memory. arXiv preprint arXiv:2110.09796, 2021.

Maja J Mataric. Reward functions for accelerated learning. In Machine learning proceedings 1994, pp. 181–189. Elsevier, 1994.

Ashvin Nair, Murtaza Dalal, Abhishek Gupta, and Sergey Levine. Accelerating online reinforcement learning with offline datasets. arXiv preprint arXiv:2006.09359, 2020.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In Icml, volume 99, pp. 278–287, 1999.

Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In Icml, volume 1, pp. 2, 2000.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. arXiv preprint arXiv:1910.00177, 2019.

Siddharth Reddy, Anca D Dragan, and Sergey Levine. Sqil: Imitation learning via reinforcement learning with sparse rewards. arXiv preprint arXiv:1905.11108, 2019.

JP Royston. Expected normal order statistics (exact and approximate): algorithm as 177. Applied Statistics, 31(2):161–5, 1982.

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. arXiv preprint arXiv:1610.03295, 2016.

Noah Y Siegel, Jost Tobias Springenberg, Felix Berkenkamp, Abbas Abdolmaleki, Michael Neunert, Thomas Lampe, Roland Hafner, Nicolas Heess, and Martin Riedmiller. Keep doing what worked: Behavioral modelling priors for offline reinforcement learning. arXiv preprint arXiv:2002.08396, 2020.

Avi Singh, Larry Yang, Chelsea Finn, and Sergey Levine. End-to-end robotic reinforcement learning without reward engineering. In Antonio Bicchi, Hadas Kress-Gazit, and Seth Hutchinson (eds.), Robotics: Science and Systems XV, University of Freiburg, Freiburg im Breisgau, Germany, June 22-26, 2019, 2019. doi: 10.15607/RSS.2019.XV.073.

Avi Singh, Albert Yu, Jonathan Yang, Jesse Zhang, Aviral Kumar, and Sergey Levine. Cog: Connecting new skills to past experience with offline reinforcement learning. arXiv preprint arXiv:2010.14500, 2020.

Shihong Song, Jiayi Weng, Hang Su, Dong Yan, Haosheng Zou, and Jun Zhu. Playing fps games with environment-aware hierarchical reinforcement learning. In IJCAI, pp. 3475–3482, 2019.

Adith Swaminathan and Thorsten Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. The Journal of Machine Learning Research, 16(1):1731–1755, 2015.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In 2012 IEEE/RSJ international conference on intelligent robots and systems, pp. 5026–5033. IEEE, 2012.

Masatoshi Uehara and Wen Sun. Pessimistic model-based offline reinforcement learning under partial coverage. arXiv preprint arXiv:2107.06226, 2021.

Yining Wang, Ruosong Wang, Simon S Du, and Akshay Krishnamurthy. Optimism in reinforcement learning with generalized linear function approximation. arXiv preprint arXiv:1912.04136, 2019.

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

Yue Wu, Shuangfei Zhai, Nitish Srivastava, Joshua Susskind, Jian Zhang, Ruslan Salakhutdinov, and Hanlin Goh. Uncertainty weighted actor-critic for offline reinforcement learning. arXiv preprint arXiv:2105.08140, 2021.

Yuxin Wu and Yuandong Tian. Training agent for first-person shooter game with actor-critic curriculum learning. In 5th International Conference on Learning Representations, ICLR 2017, 2017.

Lin Yang and Mengdi Wang. Sample-optimal parametric q-learning using linearly additive features. In International Conference on Machine Learning, pp. 6995–7004. PMLR, 2019.

Yiqin Yang, Xiaoteng Ma, Chenghao Li, Zewu Zheng, Qiyuan Zhang, Gao Huang, Jun Yang, and Qianchuan Zhao. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. arXiv preprint arXiv:2106.03400, 2021.

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, 2020a.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. arXiv preprint arXiv:2005.13239, 2020b.

Tianhe Yu, Aviral Kumar, Yevgen Chebotar, Karol Hausman, Sergey Levine, and Chelsea Finn. Conservative data sharing for multi-task offline reinforcement learning. Advances in Neural Information Processing Systems, 34:11501–11516, 2021a.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. arXiv preprint arXiv:2102.08363, 2021b.

Tianhe Yu, Aviral Kumar, Yevgen Chebotar, Karol Hausman, Chelsea Finn, and Sergey Levine. How to leverage unlabeled data in offline reinforcement learning. arXiv preprint arXiv:2202.01741, 2022.

A ALGORITHM DETAILS

In this section, we describe the details of PEVI algorithm.

In linear MDPs, we can construct $\widehat{\mathbb{B}}\widehat{V}$ and Γ based on $\mathcal{D}$ as follows, where $\widehat{\mathbb{B}}\widehat{V}$ is the empirical estimation for $\mathbb{B}\widehat{V}$. For a given dataset $\mathcal{D} = \{(s_\tau, a_\tau, r_\tau)\}_{\tau=1}^N$, we define the empirical mean squared Bellman error (MSBE) as

$$M(w) = \sum_{\tau=1}^{N} \big(r_\tau + \gamma \widehat{V}(s_{\tau+1}) - \phi(s_\tau, a_\tau)^\top w\big)^2 + \lambda ||w||_2^2$$

Here $\lambda > 0$ is the regularization parameter. Note that $\widehat{w}$ has the closed form

$$\begin{aligned} \widehat{w} &= \Lambda^{-1}\Big(\sum_{\tau=1}^{N} \phi(s_\tau, a_\tau) \cdot \big(r_\tau + \gamma \widehat{V}(s_{\tau+1})\big)\Big), \\ \text{where } \ \Lambda &= \lambda I + \sum_{\tau=1}^{N} \phi(s_\tau, a_\tau)\phi(s_\tau, a_\tau)^\top. \end{aligned} \tag{17}$$

Then we simply let

$$\widehat{\mathbb{B}}\widehat{V} = \langle \phi, \widehat{w} \rangle. \tag{18}$$

Meanwhile, we construct Γ based on $\mathcal{D}$ as

$$\Gamma(s, a) = \beta \cdot \big(\phi(s, a)^\top \Lambda^{-1} \phi(s, a)\big)^{1/2}. \tag{19}$$

Here $\beta > 0$ is the scaling parameter. The overall PEVI algorithm is summarized in Algorithm 3.

Algorithm 3 Pessimistic Value Iteration, PEVI

1: **Require**: Dataset $\mathcal{D} = \{(s_\tau, a_\tau, r_\tau, s_{\tau+1})\}_{\tau=1}^T$.
2: Initialization: Set $\widehat{V}(\cdot) \leftarrow 0$ and construct $\Gamma(\cdot, \cdot)$.
3: **while** not converged **do**
4: Construct $(\widehat{\mathbb{B}}\widehat{V})(\cdot, \cdot)$
5: Set $\widehat{Q}(\cdot, \cdot) \leftarrow (\widehat{\mathbb{B}}\widehat{V})(\cdot, \cdot) - \Gamma(\cdot, \cdot)$.
6: Set $\widehat{\pi}(\cdot \,|\, \cdot) \leftarrow \operatorname{argmax}_\pi \mathbb{E}_\pi\Big[\widehat{Q}(\cdot, \cdot)\Big]$.
7: Set $\widehat{V}(\cdot) \leftarrow \mathbb{E}_{\widehat{\pi}}\Big[\widehat{Q}(\cdot, \cdot)\Big]$.
8: **end while**
9: **Return** $\widehat{V}, \widehat{\pi}$

B PROOF OF THEOREM 4.3

Proof. From Equation (11) in Algorithm 1, we have

$$\widehat{V}_{\widetilde{\theta}} \leq \widehat{V}_\theta, \quad \forall \theta \in \mathcal{C}(\delta), \tag{20}$$

where $\widetilde{\theta}$ is the pessimistic estimation of θ.

Let $\mathcal{E}_1$ be the event $\theta^\star \in \mathcal{C}(\delta)$, then we have $\mathcal{P}(\mathcal{E}_1) \geq 1 - \delta$ from Lemma 4.1.

Let $\mathcal{E}_2$ be the event where the following inequality holds,

$$|(\mathbb{B}\widehat{V})(s, a) - (\widehat{\mathbb{B}}\widehat{V})(s, a)| \leq \Gamma = \beta\sqrt{\phi(s, a)^\top \Lambda^{-1} \phi(s, a)}, \forall (s, a) \in \mathcal{S} \times \mathcal{A}. \tag{21}$$

then we have $\mathcal{P}(\mathcal{E}_2) \geq 1 - \delta$ from Lemma C.3.

Condition on $\mathcal{E}_1 \cap \mathcal{E}_2$, we have.

$$\begin{aligned}
V_{\theta^\star}^{\pi^*} - V_{\theta^\star}^{\widehat{\pi}} &= V_{\theta^\star}^{\pi^*} - \widehat{V}_{\theta^\star} + \widehat{V}_{\theta^\star} - V_{\theta^\star}^{\widehat{\pi}} \\
&\leq V_{\theta^\star}^{\pi^*} - \widehat{V}_{\theta^\star} \\
&= V_{\theta^\star}^{\pi^*} - V_{\widetilde{\theta}}^{\pi^*} + V_{\widetilde{\theta}}^{\pi^*} - \widehat{V}_{\widetilde{\theta}} + \widehat{V}_{\widetilde{\theta}} - \widehat{V}_{\theta^\star} \\
&\leq V_{\theta^\star}^{\pi^*} - V_{\widetilde{\theta}}^{\pi^*} + V_{\widetilde{\theta}}^{\pi^*} - \widehat{V}_{\widetilde{\theta}} \\
&= V_{\theta^\star}^{\pi^*} - V_{\widehat{\theta}}^{\pi^*} + V_{\widehat{\theta}}^{\pi^*} - V_{\widetilde{\theta}}^{\pi^*} + V_{\widetilde{\theta}}^{\pi^*} - \widehat{V}_{\widetilde{\theta}} \\
&\leq \frac{4r_{\max}}{1-\gamma}\sqrt{\frac{d^2\zeta_2}{N_0 C_0^\dagger}} + \frac{2cr_{\max}}{(1-\gamma)^2}\sqrt{\frac{d^3\zeta_1}{N_0 C_0^\dagger + N_1 C_1^\dagger}},
\end{aligned}$$

where the first inequality follows from Lemma C.2. The second inequality follows from Equation (20), and the last inequality follows from Lemma C.5 and C.1.

From the union bound, we have that the above inequality holds with a probability of $1 - 2\delta$. □

C ADDTIONAL LEMMAS AND MISSING PROOFS

Lemma C.1. *Under the event in Lemma C.3, we have*

$$V_\theta^{\pi^*}(s) - \widehat{V}_\theta(s) \leq \frac{2cr_{max}}{(1-\gamma)^2}\sqrt{\frac{d^3\zeta}{C^\dagger N}},$$

with probability $1-\delta$*, for all* $\|\theta\|_2^2 \leq d$*.*

Proof. Let

$$\delta(s,a) = r(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}\widehat{V}(s') - \widehat{Q}(s,a), \tag{22}$$

From the definition of $\widehat{Q}(s,a)$ and $\widehat{V}(s)$, we have

$$\delta(s,a) = \mathbb{B}_\gamma \widehat{V}(s) - \widehat{Q}(s,a) = \mathbb{B}_\gamma \widehat{V}(s) - \widehat{\mathbb{B}}_\gamma \widehat{V} + \Gamma(s,a). \tag{23}$$

Under the condition of Lemma C.3, it holds that

$$0 \leq \delta(s,a) \leq 2\Gamma(s,a), \text{for all } s, a. \tag{24}$$

Then we have

$$\begin{aligned}
&V_\theta^{\pi^*}(s) - \widehat{V}_\theta(s) \\
=&\mathbb{E}_{a\sim\pi^*, s'\sim\mathcal{P}(\cdot|s,a)}\left[r(s,a) + \gamma V^{\pi^*}(s')\right] - \mathbb{E}_{a\sim\widehat{\pi}}\left[\widehat{Q}(s,a)\right] \\
=&\mathbb{E}_{a\sim\pi^*, s'\sim\mathcal{P}(\cdot|s,a)}\left[r(s,a) + \gamma V^{\pi^*}(s') - \widehat{Q}(s,a)\right] + \mathbb{E}_{a\sim\pi^*}\left[\widehat{Q}(s,a)\right] - \mathbb{E}_{a\sim\widehat{\pi}}\left[\widehat{Q}(s,a)\right] \\
=&\mathbb{E}_{a\sim\pi^*, s'\sim\mathcal{P}(\cdot|s,a)}\left[r(s,a) + \gamma \widehat{V}(s') - \widehat{Q}(s,a)\right] + \gamma\mathbb{E}_{a\sim\pi^*, s'\sim\mathcal{P}(\cdot|s,a)}\left[V^{\pi^*}(s') - \widehat{V}(s')\right] \\
&+ \left\langle \widehat{Q}(s,\cdot), \pi^*(\cdot\,|\,s) - \widehat{\pi}(\cdot\,|\,s)\right\rangle_{\mathcal{A}} \\
=&\mathbb{E}_{a\sim\pi^*, s'\sim\mathcal{P}(\cdot|s,a)}\left[\delta(s,a)\right] + \left\langle \widehat{Q}(s,\cdot), \pi^*(\cdot\,|\,s) - \widehat{\pi}(\cdot\,|\,s)\right\rangle_{\mathcal{A}} + \cdots \\
=&\mathbb{E}_{\pi^*}\left[\sum_{t=0}^{\infty}\gamma^t\delta(s_t,a_t)\,|\,s_0 = s\right] + \mathbb{E}_{\pi^*}\left[\sum_{t=0}^{\infty}\gamma^t\left\langle \widehat{Q}(s_t,\cdot), \pi^*(\cdot\,|\,s_t) - \widehat{\pi}(\cdot\,|\,s_t)\right\rangle_{\mathcal{A}}\,|\,s_0 = s\right] \\
\leq&\mathbb{E}_{\pi^*}\left[\sum_{t=0}^{\infty}\gamma^t\delta(s_t,a_t)\,\middle|\,s_0 = s\right] \\
\leq&2\mathbb{E}_{\pi^*}\left[\sum_{t=0}^{\infty}\gamma^t\Gamma(s_t,a_t)\,\middle|\,s_0 = s\right] \\
=&2\beta\mathbb{E}_{\pi^*}\left[\sum_{t=0}^{\infty}\gamma^t\left(\phi(s_t,a_t)^\top\Lambda^{-1}\phi(s_t,a_t)\right)^{1/2}\,\middle|\,s_0 = s\right].
\end{aligned}$$

Here the first inequality follows from the fact that $\widehat{\pi}(\cdot|s) = \operatorname{argmax}_\pi \left\langle \widehat{Q}(\cdot,\cdot), \pi(\cdot|s) \right\rangle$ and the second inequality follows from Equation (24).

By the Cauchy-Schwarz inequality, we have

$$\begin{aligned}
&\mathbb{E}_{\pi^*}\Big[\sum_{t=0}^{\infty}\gamma^t\big(\phi(s_t,a_t)^\top\Lambda^{-1}\phi(s_t,a_t)\big)^{1/2}\,\Big|\,s_0=s\Big]\\
&\quad=\frac{1}{1-\gamma}\mathbb{E}_{d^{\pi^*}}\Big[\sqrt{\mathrm{Tr}\left(\phi(s,a)^\top\Lambda^{-1}\phi(s,a)\right)}\,\Big|\,s_0=s\Big]\\
&\quad=\frac{1}{1-\gamma}\mathbb{E}_{d^{\pi^*}}\Big[\sqrt{\mathrm{Tr}\left(\phi(s,a)\phi(s,a)^\top\Lambda^{-1}\right)}\,\Big|\,s_0=s\Big]\\
&\quad\le\frac{1}{1-\gamma}\sqrt{\mathrm{Tr}\left(\mathbb{E}_{d^{\pi^*}}\big[\phi(s,a)\phi(s,a)^\top\,\big|\,s_0=s\big]\Lambda^{-1}\right)}\\
&\quad=\frac{1}{1-\gamma}\sqrt{\mathrm{Tr}\left(\Sigma_{\pi^*,s}^\top\Lambda^{-1}\right)},
\end{aligned}\tag{25}$$

for all $s \in \mathcal{S}$. Then we have

$$\begin{aligned}
V_\theta^{\pi^*}(s)-\widehat{V}_\theta(s)&\le 2\beta\mathbb{E}_{\pi^*}\Big[\sum_{t=0}^{\infty}\gamma^t\big(\phi(s_t,a_t)^\top\Lambda^{-1}\phi(s_t,a_t)\big)^{1/2}\,\Big|\,s_0=s\Big]\\
&\le\frac{2\beta}{1-\gamma}\sqrt{\mathrm{Tr}\left(\Sigma_{\pi^*,s}\cdot\big(I+C^\dagger\cdot N\cdot\Sigma_{\pi^*,s}\big)^{-1}\right)}\\
&=\frac{2\beta}{1-\gamma}\sqrt{\sum_{j=1}^{d}\frac{\lambda_j(s)}{1+C^\dagger\cdot N\cdot\lambda_j(s)}}.
\end{aligned}\tag{26}$$

Here $\{\lambda_j(s)\}_{j=1}^d$ are the eigenvalues of $\Sigma_{\pi^*,s}$ for all $s \in \mathcal{S}$, the second inequality follows from Equation (25). Meanwhile, by Definition 3.1, we have $\|\phi(s,a)\| \le 1$ for all $(s,a) \in \mathcal{S}\times\mathcal{A}$. By Jensen's inequality, we have

$$\|\Sigma_{\pi^*,s}\|_{\mathrm{op}} \le \mathbb{E}_{\pi^*}\big[\|\phi(s,a)\phi(s,a)^\top\|_{\mathrm{op}}\,\big|\,s_0=s\big] \le 1 \tag{27}$$

for all $s \in \mathcal{S}$. As $\Sigma_{\pi^*,s}$ is positive semidefinite, we have $\lambda_j(s) \in [0,1]$ for all $s \in \mathcal{S}$ and all $j \in [d]$. Hence we have

$$\begin{aligned}
V_\theta^{\pi^*}(s)-\widehat{V}_\theta(s)&\le\frac{2\beta}{1-\gamma}\sqrt{\sum_{j=1}^{d}\frac{\lambda_j(s)}{1+C^\dagger\cdot N\cdot\lambda_j(s)}}\\
&\le\frac{2\beta}{1-\gamma}\sqrt{\sum_{j=1}^{d}\frac{1}{1+C^\dagger\cdot N}}\le\frac{2cr_{\max}}{(1-\gamma)^2}\sqrt{\frac{d^3\zeta}{C^\dagger N}}
\end{aligned}\tag{28}$$

for all $x \in \mathcal{S}$, where the second inequality follows from the fact that $\lambda_j(s) \in [0,1]$ for all $s \in \mathcal{S}$ and all $j \in [d]$, while the third inequality follows from the choice of the scaling parameter $\beta > 0$.

Then we have the conclusion in Lemma C.1. □

Lemma C.2. *Under the event in Lemma C.3, we have*

$$\widehat{V}_\theta(s) - V^{\widehat{\pi}_\theta}(s) \le 0 \tag{29}$$

with probability $1-\delta$*, for all* $\|\theta\|_2^2 \le d$*.*

Proof. Similar to the proof of Lemma C.1, let

$$\delta(s,a) = r(s,a) + \gamma\mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}\widehat{V}(s') - \widehat{Q}(s,a), \tag{30}$$

we have

$$\begin{aligned}\widehat{V}(s) - V^{\widehat{\pi}}(s) =& \mathbb{E}_{a\sim\widehat{\pi}}\left[\widehat{Q}(s,a)\right] - \mathbb{E}_{a\sim\widehat{\pi},s'\sim\mathcal{P}(\cdot|s,a)}\left[r(s,a)+\gamma V^{\widehat{\pi}}(s')\right]\\ =& \mathbb{E}_{a\sim\widehat{\pi},s'\sim\mathcal{P}(\cdot|s,a)}\left[\widehat{Q}(s,a) - r(s,a) - \gamma\widehat{V}(s')\right]\\ &+\gamma\mathbb{E}_{a\sim\widehat{\pi},s'\sim\mathcal{P}(\cdot|s,a)}\left[\widehat{V}(s') - V^{\widehat{\pi}}(s')\right]\\ =& -\mathbb{E}_{\widehat{\pi}}\left[\delta(s,a)\right] + \gamma\mathbb{E}_{a\sim\widehat{\pi},s'\sim\mathcal{P}(\cdot|s,a)}\left[\widehat{V}(s') - V^{\widehat{\pi}}(s')\right]\\ =& -\mathbb{E}_{\widehat{\pi}}\left[\delta(s,a)\right] + \cdots\\ =& -\mathbb{E}_{\widehat{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t\delta(s_t,a_t)\mid s_0=s\right].\end{aligned}$$

Then under the condition of Lemma C.3, it holds that

$$0 \le \delta(s,a) \le 2\Gamma(s,a), \text{for all } s,a, \tag{31}$$

Then we have the result immediately.

□

Lemma C.3 (ξ-Quantifiers)**.** *Let*

$$\lambda = 1, \quad \beta = c\cdot dV_{max}\sqrt{\zeta}, \quad \zeta = \log\left(2dN/(1-\gamma)\xi\right). \tag{32}$$

Then $\Gamma = \beta\cdot\left(\phi(s,a)^\top\Lambda^{-1}\phi(s,a)\right)^{1/2}$ *are* ξ*-quantifiers with probability at least* $1-\xi$*. That is, let* $\mathcal{E}_2$ *be the event that the following inequality holds,*

$$|(\mathbb{B}\widehat{V})(s,a) - (\widehat{\mathbb{B}}\widehat{V})(s,a)| \le \Gamma = \beta\sqrt{\phi(s,a)^\top\Lambda^{-1}\phi(s,a)}, \forall(s,a)\in\mathcal{S}\times\mathcal{A}. \tag{33}$$

Then we have $\mathcal{P}(\mathcal{E}_2)\ge 1-\varepsilon$*.*

Proof. we have

$$\begin{aligned}\mathbb{B}\widehat{V} - \widehat{\mathbb{B}}\widehat{V} &= \phi(s,a)^\top(w-\widehat{w})\\ &= \phi(s,a)^\top w - \phi(s,a)\Lambda^{-1}\left(\sum_{\tau=1}^{N}\phi_\tau(r_\tau+\gamma\widehat{V}(s_{\tau+1})\right)\\ &= \underbrace{\phi(s,a)^\top w - \phi(s,a)\Lambda^{-1}\left(\sum_{\tau=1}^{N}\phi_\tau\phi_\tau^\top w\right)}_{\text{(i)}} + \underbrace{\phi(s,a)\Lambda^{-1}(\sum_{\tau=1}^{N}\phi_\tau\phi_\tau^\top w - \sum_{\tau=1}^{N}\phi_\tau(r_\tau+\gamma\widehat{V}(s_{\tau+1})),}_{\text{(ii)}}\end{aligned} \tag{34}$$

Then we bound (i) and (ii), respectively.

For (i), we have

$$\begin{aligned}\text{(i)} &= \phi(s,a)^\top w - \phi(s,a)\Lambda^{-1}(\Lambda-\lambda I)w\\ &= \lambda\phi(s,a)\Lambda^{-1}w\\ &\le \lambda||\phi(s,a)||_{\lambda^{-1}}||w||_{\lambda^{-1}}\\ &\le V_{\max}\sqrt{d\lambda}\sqrt{\phi(s,a)^\top\Lambda^{-1}\phi(s,a)},\end{aligned} \tag{35}$$

where the first inequality follows from Cauchy-Schwartz inequality. The second inequality follows from the fact that $||\Lambda^{-1}||_{\text{op}} \le \lambda^{-1}$ and Lemma C.4.

For notation simplicity, let $\epsilon_\tau = r_\tau + \gamma \widehat{V}(s_{\tau+1}) - \phi_\tau^\top w$, then we have

$$\begin{aligned}
|\text{(ii)}| &= \phi(s,a)\Lambda^{-1}\sum_{\tau=1}^N \phi_\tau \epsilon_\tau \\
&\leq ||\sum_{\tau=1}^N \phi_\tau \epsilon_\tau||_{\Lambda^{-1}} \cdot ||\phi(s,a)||_{\Lambda^{-1}} \\
&= \underbrace{||\sum_{\tau=1}^N \phi_\tau \epsilon_\tau||_{\Lambda^{-1}}}_{\text{(iii)}} \cdot \sqrt{\phi(s,a)^\top \Lambda^{-1}\phi(s,a)}. \qquad (36)
\end{aligned}$$

The term (iii) is depend on the randomness of the data collection process of $\mathcal{D}$. To bound this term, we resort to uniform concentration inequalities to upper bound

$$\sup_{V\in\mathcal{V}(R,B,\lambda)} \Big\| \sum_{\tau=1}^N \phi(s_\tau, a_\tau)\cdot \epsilon_\tau(V)\Big\|,$$

where

$$\mathcal{V}(R,B,\lambda) = \{V(s;w,\beta,\Sigma) : \mathcal{S}\to[0,V_{\max}] \text{ with} ||w|| \leq R, \beta\in[0,B], \Sigma \succeq \lambda\cdot I\}, \qquad (37)$$

where $V(s;w,\beta,\Sigma) = \max_a\{\phi(s,a)^\top w - \beta\cdot\sqrt{\phi(s,a)^\top\Sigma^{-1}\phi(s,a)}\}$. For all $\epsilon > 0$, let $\mathcal{N}(\epsilon;R,B,\lambda)$ be the minimal cover if $\mathcal{V}(R,B,\lambda)$. That is, for any function $V\in\mathcal{V}(R,B,\lambda)$, there exists a function $V^\dagger\in\mathcal{N}(\epsilon;R,B,\lambda)$, such that

$$\sup_{s\in\mathcal{S}} |V(s) - V^\dagger(s)| \leq \epsilon. \qquad (38)$$

Let $R_0 = V_{\max}\sqrt{Nd/\lambda}, B_0 = 2\beta$, it is easy to show that at each iteration, $\widehat{V}^u \in \mathcal{V}(R_0,B_0,\lambda)$. From the definition of $\mathbb{B}$, we have

$$|\mathbb{B}\widehat{V} - \mathbb{B}V^\dagger| = \gamma\left|\int (\widehat{V}(s') - V^\dagger(s'))\,\langle\phi(s,a),\mu(s')\rangle\,\mathrm{d}s'\right| \leq \gamma\epsilon. \qquad (39)$$

Then we have

$$|(r+\gamma V - \mathbb{B}V) - (r+\gamma V^\dagger - \mathbb{B}V^\dagger)| \leq 2\gamma\epsilon. \qquad (40)$$

Let $\epsilon_\tau^\dagger = r(s_\tau,a_\tau) + \gamma V^\dagger(s_{\tau+1}) - \mathbb{B}V^\dagger(s,a)$, we have

$$\begin{aligned}
\text{(iii)}^2 = ||\sum_{\tau=1}^N \phi_\tau\epsilon_\tau||_{\Lambda^{-1}}^2 &\leq 2||\sum_{\tau=1}^N \phi_\tau\epsilon_\tau^\dagger||_{\Lambda^{-1}}^2 + 2||\sum_{\tau=1}^N \phi_\tau(\epsilon_\tau^\dagger - \epsilon_\tau)||_{\Lambda^{-1}}^2 \\
&\leq 2||\sum_{\tau=1}^N \phi_\tau\epsilon_\tau^\dagger||_{\Lambda^{-1}}^2 + 8\gamma^2\epsilon^2\sum_{\tau=1}^N |\phi_\tau\Lambda^{-1}\phi_\tau| \\
&\leq 2||\sum_{\tau=1}^N \phi_\tau\epsilon_\tau^\dagger||_{\Lambda^{-1}}^2 + 8\gamma^2\epsilon^2 N^2/\lambda
\end{aligned}$$

It remains to bound $||\sum_{\tau=1}^N \phi_\tau\epsilon_\tau^\dagger||_{\Lambda^{-1}}^2$. From the assumption for data collection process, it is easy to show that $\mathbb{E}_{\mathcal{D}}[\epsilon_\tau \,|\, \mathcal{F}_{\tau-1}] = 0$, where $F_{\tau-1} = \sigma(\{(s_i,a_i)_{i=1}^\tau \cup (r_i,s_{i+1})_{i=1}^\tau\})$ is the σ-algebra generated by the variables from the first τ step. Moreover, since $\epsilon_\tau \leq 2V_{\max}$, we have ϵ_τ are $2V_{\max}$-sub-Gaussian conditioning on $F_{\tau-1}$. Then we invoke Lemma C.7 with $M_0 = \lambda\cdot I$ and $M_k = \lambda\cdot I + \sum_{\tau=1}^k \phi(s_\tau,a_\tau)\,\phi(s_\tau,a_\tau)^\top$. For the fixed function $V\colon\mathcal{S}\to[0,V_{\max}]$, we have

$$\mathbb{P}_{\mathcal{D}}\left(\Big\|\sum_{\tau=1}^N \phi(s_\tau,a_\tau)\cdot\epsilon_\tau(V)\Big\|_{\Lambda^{-1}}^2 > 8V_{\max}^2\cdot\log\Big(\frac{\det(\Lambda)^{1/2}}{\delta\cdot\det(\lambda\cdot I)^{1/2}}\Big)\right) \leq \delta \qquad (41)$$

for all $\delta\in(0,1)$. Note that $\|\phi(s,a)\|\leq 1$ for all $(s,a)\in\mathcal{S}\times\mathcal{A}$ by Definition 3.1. We have

$$\|\Lambda\|_{\text{op}} = \Big\|\lambda\cdot I + \sum_{\tau=1}^N \phi(s_\tau,a_\tau)\phi(s_\tau,a_\tau)^\top\Big\|_{\text{op}} \leq \lambda + \sum_{\tau=1}^N \|\phi(s_\tau,a_\tau)\phi(s_\tau,a_\tau)^\top\|_{\text{op}} \leq \lambda + N,$$

where $\|\cdot\|_{\text{op}}$ denotes the matrix operator norm. Hence, it holds that $\det(\Lambda) \leq (\lambda + N)^d$ and $\det(\lambda \cdot I) = \lambda^d$, which implies

$$\mathbb{P}_{\mathcal{D}}\bigg(\Big\|\sum_{\tau=1}^{N}\phi(s_\tau, a_\tau)\cdot\epsilon_\tau(V)\Big\|_{\Lambda_{-1}}^2 > 4V_{\max}^2\cdot\big(2\cdot\log(1/\delta) + d\cdot\log(1+N/\lambda)\big)\bigg)$$
$$\leq \mathbb{P}_{\mathcal{D}}\bigg(\Big\|\sum_{\tau=1}^{N}\phi(s_\tau, a_\tau)\cdot\epsilon_\tau(V)\Big\|_{\Lambda_{-1}}^2 > 8V_{\max}^2\cdot\log\Big(\frac{\det(\Lambda)^{1/2}}{\delta\cdot\det(\lambda\cdot I)^{1/2}}\Big)\bigg) \leq \delta.$$

Therefore, we conclude the proof of Lemma C.3.

Applying Lemma C.3 and the union bound, we have

$$\mathbb{P}_{\mathcal{D}}\bigg(\sup_{V\in\mathcal{N}(\varepsilon)}\Big\|\sum_{\tau=1}^{N}\phi(s_\tau, a_\tau)\cdot\epsilon_\tau(V)\Big\|_{\Lambda^{-1}}^2 > 4V_{\max}^2\cdot\big(2\cdot\log(1/\delta)+d\cdot\log(1+N/\lambda)\big)\bigg) \leq \delta\cdot|\mathcal{N}(\varepsilon)|. \tag{42}$$

Recall that

$$\widehat{V} \in \mathcal{V}(R_0, B_0, \lambda), \qquad \text{where } R_0 = V_{\max}\sqrt{Nd/\lambda},\ B_0 = 2\beta,\ \lambda = 1,\ \beta = c\cdot dV_{\max}\sqrt{\zeta}. \tag{43}$$

Here $c > 0$ is an absolute constant, $\xi \in (0,1)$ is the confidence parameter, and $\zeta = \log(2dV_{\max}/\xi)$ is specified in Algorithm 3. Applying Lemma C.6 with $\varepsilon = dV_{\max}/N$, we have

$$\log|\mathcal{N}(\varepsilon)| \leq d\cdot\log(1+4d^{-1/2}N^{3/2}) + d^2\cdot\log(1+32c^2\cdot d^{1/2}N^2\zeta)$$
$$\leq d\cdot\log(1+4d^{1/2}N^2) + d^2\cdot\log(1+32c^2\cdot d^{1/2}N^2\zeta). \tag{44}$$

By setting $\delta = \xi/|\mathcal{N}(\varepsilon)|$, we have that with probability at least $1-\xi$,

$$\Big\|\sum_{\tau=1}^{N}\phi(s_\tau, a_\tau)\cdot\epsilon_\tau(\widehat{V})\Big\|_{\Lambda^{-1}}^2$$
$$\leq 8V_{\max}^2\cdot\big(2\cdot\log(V_{\max}/\xi) + 4d^2\cdot\log(64c^2\cdot d^{1/2}N^2\zeta) + d\cdot\log(1+N) + 4d^2\big)$$
$$\leq 8V_{\max}^2 d^2\zeta(4+\log(64c^2)). \tag{45}$$

Here the last inequality follows from simple algebraic inequalities. We set $c \geq 1$ to be sufficiently large, which ensures that $36 + 8\cdot\log(64c^2) \leq c^2/4$ on the right-hand side of Equation (45). By Equations (36) and (45), it holds that

$$|\text{(ii)}| \leq c/2\cdot dV_{\max}\sqrt{\zeta}\cdot\sqrt{\phi(x,a)^\top\Lambda^{-1}\phi(s,a)} = \beta/2\cdot\sqrt{\phi(x,a)^\top\Lambda^{-1}\phi(s,a)} \tag{46}$$

By Equations (19), (34), (35), and (46), for all $(s,a) \in \mathcal{S}\times\mathcal{A}$, it holds that

$$\big|(\mathbb{B}\widehat{V})(s,a) - (\widehat{\mathbb{B}}\widehat{V})(s,a)\big| \leq (V_{\max}\sqrt{d} + \beta/2)\cdot\sqrt{\phi(s,a)^\top\Lambda^{-1}\phi(s,a)} \leq \Gamma(s,a) \tag{47}$$

with probability at least $1-\xi$. Therefore, we conclude the proof of Lemma C.3. □

Lemma C.4 (Bounded weight of value function)**.** *Let $V_{\max} = r_{\max}/(1-\gamma)$. For any function $V : \mathcal{S} \to [0, V_{\max}]$, we have*

$$||w|| \leq V_{\max}\sqrt{d}, ||\widehat{w}|| \leq V_{\max}\sqrt{\frac{Nd}{\lambda}}.$$

Proof. Since

$$w^\top\phi(s,a) = \langle M, \phi(s,a)\rangle + \gamma\int V(s')\psi(s')^\top M\phi(s,a)\mathrm{d}s',$$

we have

$$\begin{aligned} w &= M + \gamma \int V(s')\psi(s')^\top M \mathrm{d}s' \\ &= r_{\max}\sqrt{d} + \gamma V_{\max}\sqrt{d} \\ &= V_{\max}\sqrt{d}. \end{aligned}$$

For $\widehat{w}$, we have

$$\begin{aligned} ||\widehat{w}|| &= ||\Lambda^{-1}\sum_{\tau=1}^{N}\phi_\tau(r_\tau + \gamma V(s_{\tau+1}))|| \\ &\leq \sum_{\tau=1}^{N}||\Lambda^{-1}\phi_\tau(r_\tau + \gamma V(s_{\tau+1}))|| \\ &\leq V_{\max}\sum_{\tau=1}^{N}||\Lambda^{-1}\phi_\tau|| \\ &\leq V_{\max}\sum_{\tau=1}^{N}\sqrt{\phi_\tau^\top\Lambda^{-1/2}\Lambda^{-1}\Lambda^{-1/2}\phi_\tau} \\ &\leq \frac{V_{\max}}{\sqrt{\lambda}}\sum_{\tau=1}^{N}\sqrt{\phi_\tau^\top\Lambda^{-1}\phi_\tau} \\ &\leq V_{\max}\sqrt{\frac{N}{\lambda}}\sqrt{\mathrm{Tr}(\Lambda^{-1}\sum_{\tau=1}^{T}\phi_\tau\phi_\tau^\top)} \\ &\leq V_{\max}\sqrt{\frac{Nd}{\lambda}}. \end{aligned}$$

□

Lemma C.5. *For policy π^* and any reward function parameter $\theta \in \mathcal{C}(\delta)$, we have*

$$|V_\theta^{\pi^*} - V_{\widehat{\theta}}^{\pi^*}| \leq \frac{2r_{max}}{1-\gamma}\sqrt{\frac{d^2\zeta_2}{N_0 C_0^\dagger}}.$$

Proof. From the definition, we have

$$\begin{aligned} |r_\theta(s,a) - r_{\widehat{\theta}}(s,a)| &= |\phi(s,a)^\top\theta - \phi(s,a)^\top\widehat{\theta}| \\ &\leq \|\theta - \widehat{\theta}\|_\Lambda \cdot \|\phi(s,a)\|_{\Lambda^{-1}} \\ &\leq \alpha\sqrt{\phi(s,a)^\top\Lambda^{-1}\phi(s,a)}, \end{aligned}$$

Where the first inequality follows from the Cauchy-Schwartz inequality and the second inequality uses the fact that $\theta \in \mathcal{C}(\delta)$. Then we have

$$\begin{aligned} V_\theta^{\pi^*}(s) - V_{\widehat{\theta}}^{\pi^*}(s) &= \mathbb{E}_{\pi^*}\left[\sum_{t=0}^{\infty} \gamma^t (r_\theta(s,a) - r_{\widehat{\theta}}(s,a)) \,\Big|\, s_0 = s\right] \\ &\le \mathbb{E}_{\pi^*}\left[\sum_{t=0}^{\infty} \gamma^t |r_\theta(s,a) - r_{\widehat{\theta}}(s,a)| \,\Big|\, s_0 = s\right] \\ &\le \alpha \mathbb{E}_{\pi^*}\left[\sum_{t=0}^{\infty} \gamma^t \sqrt{\phi(s,a)^\top \Lambda^{-1} \phi(s,a)} \,\Big|\, s_0 = s\right] \\ &\le \frac{\alpha}{1-\gamma}\sqrt{\sum_{j=1}^{d} \frac{1}{1 + C_0^\dagger \cdot N_0}} \le \frac{2r_{\max}}{1-\gamma}\sqrt{\frac{d^2\zeta_2}{N_0 C_0^\dagger}}, \end{aligned}$$

where the second to last inequality follows similarly as Equation (25) (26) and the last inequality follows from the fact that $\alpha = 2r_{\max}\sqrt{d\zeta_2}$.

Note that such a choice of α is sufficient for Lemma 4.1 to hold since

$$\begin{aligned} \sqrt{\nu} + r_{\max} \cdot \sqrt{2\log\frac{1}{\delta} + d\log(1 + \frac{N_0}{\nu d})} &= 1 + r_{\max} \cdot \sqrt{2\log\frac{1}{\delta} + d\log(1 + \frac{N_0}{d})} \\ &\le 1 + r_{\max} \cdot \sqrt{2\log\frac{1}{\delta} + d\log 2N_0} \\ &\le 1 + r_{\max} \cdot \sqrt{d\log\frac{2N_0}{\delta}} \\ &\le 2r_{\max} \cdot \sqrt{d\log\frac{2N_0}{\delta}} \\ &= 2r_{\max}\sqrt{d\zeta_2}, \end{aligned}$$

where the inequalities holds for sufficiently small $\delta > 0$ and $d \ge 2$. □

C.1 Technical Lemmas

Lemma C.6 (ε-Covering Number (Jin et al., 2020))**.** *For all $h \in [H]$ and all $\varepsilon > 0$, we have*

$$\log|\mathcal{N}(\varepsilon; R, B, \lambda)| \le d \cdot \log(1 + 4R/\varepsilon) + d^2 \cdot \log\big(1 + 8d^{1/2}B^2/(\varepsilon^2\lambda)\big).$$

Proof of Lemma C.6. See Lemma D.6 in Jin et al. (2020) for detailed proof. □

Lemma C.7 (Concentration of Self-Normalized Processes (Abbasi-Yadkori et al., 2011))**.** *Let $\{\mathcal{F}_t\}_{t=0}^\infty$ be a filtration and $\{\epsilon_t\}_{t=1}^\infty$ be an $\mathbb{R}$-valued stochastic process such that ϵ_t is $\mathcal{F}_t$-measurable for all $t \ge 1$. Moreover, suppose that conditioning on $\mathcal{F}_{t-1}$, ϵ_t is a zero-mean and σ-sub-Gaussian random variable for all $t \ge 1$, that is,*

$$\mathbb{E}[\epsilon_t \,|\, \mathcal{F}_{t-1}] = 0, \qquad \mathbb{E}\big[\exp(\lambda\epsilon_t) \,\big|\, \mathcal{F}_{t-1}\big] \le \exp(\lambda^2\sigma^2/2), \qquad \forall \lambda \in \mathbb{R}.$$

Meanwhile, let $\{\phi_t\}_{t=1}^\infty$ be an $\mathbb{R}^d$-valued stochastic process such that ϕ_t is $\mathcal{F}_{t-1}$-measurable for all $t \ge 1$. Also, let $M_0 \in \mathbb{R}^{d\times d}$ be a deterministic positive-definite matrix and

$$M_t = M_0 + \sum_{s=1}^{t} \phi_s \phi_s^\top$$

for all $t \ge 1$. For all $\delta > 0$, it holds that

$$\Big\|\sum_{s=1}^{t} \phi_s \epsilon_s\Big\|_{M_t^{-1}}^2 \le 2\sigma^2 \cdot \log\Big(\frac{\det(M_t)^{1/2} \cdot \det(M_0)^{-1/2}}{\delta}\Big)$$

for all $t \ge 1$ with probability at least $1 - \delta$.

Proof. See Theorem 1 of Abbasi-Yadkori et al. (2011) for a detailed proof. □

D CONSTRUCTION OF ADVERSARIAL EXAMPLES

In this section, we show that there is an MDP and an "unfortunate" dataset such that the suboptimality of Algorithm 3 matches the performance bound in Theorem 4.3. We first focus on the case without data sharing. The same techniques can be used to match the reward-learning bound. We only need to show that all inequalities in Theorem 4.3 can become equality. Suppose we have an MDP with one state and $N > d$ actions. d of them are optimal actions, with feature map

$$\phi(s, a_i) = (\underbrace{0, \dots}_{i-1 \text{ zeros}}, \sqrt{d}, 0, \dots)$$

and let the optimal policy be uniform over d actions. Such construction makes $\Sigma_{\pi*,s} = I$ so that inequalities in Equation (25)∼(27) become equalities. Then we let samples from other actions match the confidence upper bound while samples from the optimal action match the confidence lower bound, and we also need all the samples to be symmetric over different dimensions (this is required by the Cauchy-Schwitz inequality used in the proof), such that the confidence bound inequalities in Lemma C.3 also become equalities. Then, in this case, the suboptimality bound is matched and the bound in Theorem 4.3 is tight.

E EXPERIMENTAL SETTING

E.1 MULTI-TASK ANTMAZE

We first divide the source dataset (e.g., antmaze-medium-diverse-v2) into the `directed` dataset or the `undirected` dataset. The trajectories in the `undirected` dataset are randomly and uniformly divided into different tasks. In contrast, the `directed` dataset is associated with the goal closest to the final state of the trajectory. Then, for each subtask (e.g., goal=(0.0, 16.0)), we relabel the corresponding `undirected` or `undirected` dataset according to the goal. Finally, we keep the rewards in the target task dataset (labeled dataset) while setting the rewards in the other task dataset to 0 (unlabeled dataset). We visualize the `directed` or `undirected` dataset in Figure 3. We can find that the distribution shift issue in the `directed` dataset is more severe than the `undirected` dataset, which further challenges the unlabeled data sharing algorithms.

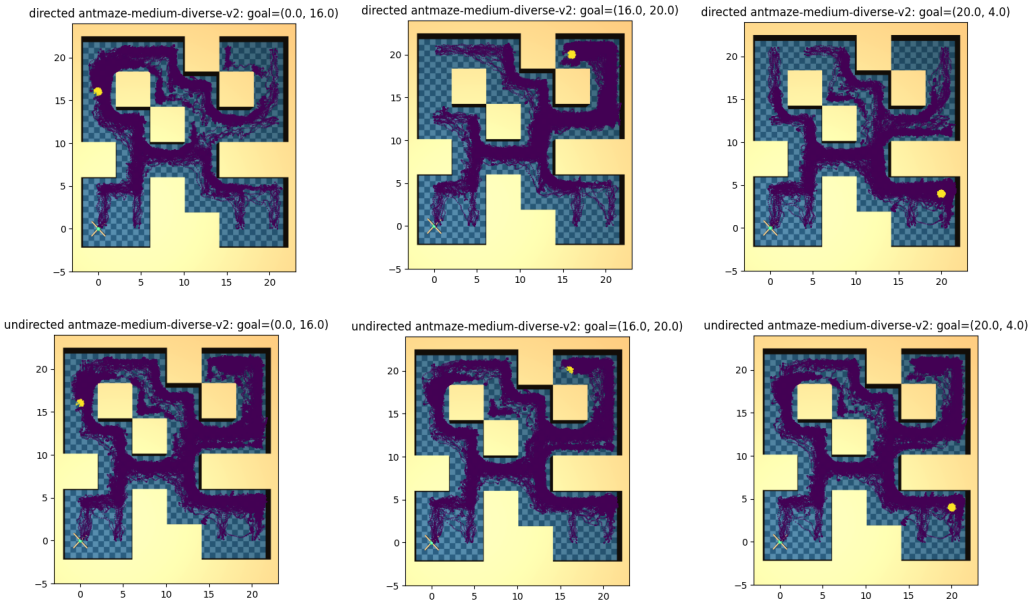
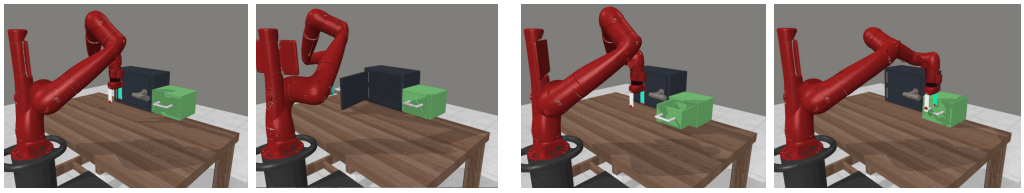


Figure 3: Visualization of multi-task antmaze-medium-diverse-v2 datasets. The purple dots denote the transition of reward 0. The yellow dots denote the transition near the goal of the sub-task and the reward is +1.

E.2 MULTI-TASK META-WORLD

We consider the same setup as in CDS (Yu et al., 2021a) and four tasks: `door open`, `door close`, `drawer open` and `drawer close`, which is shown in Figure 4. We use medium-replay datasets with 152K transitions for task door open and drawer close and use 10 expert trajectories for task door close and drawer open.



(a) Door Open (b) Drawer Close

Figure 4: Visualization of subtasks in Meta-World.

F MORE DISCUSSION ON THE RELATIVE PERFORMANCE OF UDS

Based on the discussion in the main text, we can summarize the factors that affect the relative performance of PDS algorithms as follows.

scenarios	large number of unlabeled data	high quality of unlabeled data	long horizon	large dimension
finite-sample term	✓	✓		
asymptotic term			✓	✓

Table 4: Scenarios where PDS has better relative performance.

However, it is worth noting that the asymptotic term is dependent on the backbone offline algorithm we choose. For model-based algorithms like Uehara & Sun (2021), the performance bound scale as $\mathcal{O}(d)$ and thus the problem's dimension does not affect the asymptotic term. The dependence over the discount factor is also dependent on the choice of the backbone algorithm. However, it is known that the lower bound of offline RL algorithms in linear MDPs scales as $\mathcal{O}((1-\gamma)^{1.5})$ so this asymptotic term must scales at least as $\mathcal{O}((1-\gamma)^{0.5})$ and thus negatively depends on the discount factor. That is, the larger the discount factor, the better the relative performance of PDS.

G DISCUSSION AND IMPROVEMENT ABOUT THE PARAMETER k

G.1 HOW TO SELECT THE PARAMETER k?

PDS is the generalization of UDS and prediction. We can improve the performance of PDS by selecting the suitable parameter k. We give the guidance for selecting k as follows:

1. Suppose the quality of the labeled and unlabeled data is similar (e.g., both labeled and unlabeled data come from the Walker2d/Hopper-medium datasets or Antmaze-Directed/Undirected datasets). In that case, we find the uncertainty of the reward prediction $\sigma(s, a)$ is low. Therefore, we select a smaller k (e.g., $k = 5$).
2. On the other hand, suppose the quality of the labeled and unlabeled data differs significantly. In that case, the uncertainty of the reward prediction is high, and we will select k according to the quality of the unlabeled data.

   Specifically, suppose the labeled data is higher quality than the unlabeled data (e.g., labeled data comes from the Walker2d-Expert datasets while the unlabeled data comes from the Walker2d-Random datasets). We select a higher k to penalize the prediction uncertainty (e.g., $k = 15$). On the contrary, suppose the labeled data is lower quality than the unlabeled data (e.g., labeled data comes from the Walker2d-Random datasets while the unlabeled data comes from the Walker2d-Expert datasets). We select a lower k (e.g., $k = 10$) since we prefer the high reward in the high-quality unlabeled data. We conduct ablation studies in Figure 1 to verify the above statement.

G.2 IS THERE EXISTS A UNIVERSAL CONSTANT THAT WORKS ACROSS DIFFERENT DOMAINS?

Although we have the guidance about k in the subsection G.1, we need to select the appropriate k for each domain, which is somewhat cumbersome. To solve this issue, we propose a simple yet efficient automatically parameter-adjustment mechanism.

We first define the reward mean of the labeled data as $\mu = \frac{1}{N0}\sum_{i=1}^{N0}\mu(s_i, a_i)$, and the prediction reward mean is denoted as $\widehat{\mu} = \frac{1}{N1}\sum_{i=1}^{N1}\widehat{\mu}(s_i, a_i)$. We observe that the quality-difference between the labeled and unlabeled datasets, is highly susceptible to the scale of $\mu - \widehat{\mu}$. Specifically, compared with the labeled data, if the quality of the unlabeled data is poor, $\widehat{\mu}$ is small and $\mu - \widehat{\mu} > 0$. On the contrary, $\mu - \widehat{\mu} < 0$ if the unlabeled data is high quality and the labeled data is low quality.

Based on the above observation, we adopt a linear function related to $\mu - \widehat{\mu}$ to measure the quality of the unlabeled data. Next, we use the standard deviation $\sigma(s, a)$ to capture the uncertainty. In summary, we adopt the following function to automatically adjust k:

$$\widehat{r}(s, a) = \max\left\{\min_{j=1,\dots,L} f_{\theta_j}(s, a) - k\sigma(s, a), 0\right\},$$
$$\text{where} \quad k = a \cdot \frac{\max(\mu - \widehat{\mu}, 0)}{|\mu| + \epsilon}.$$

Here we use the minimum over L ensembles rather than the mean value to impose the additional pessimistic penalties. We use $a = 25$ and $L = 10$ in all experiments. The experimental results in Table 5 demonstrate that the PDS with the automatic parameter-adjustment mechanism achieves strong performance across various domains. The experimental results also show that there is a universal parameter a to adaptively adjust k that generally works better than the baselines.

Env	UDS	Prediction	PDS++
Walker2d-medium/50K-medium/0.1M	71.0±3.9	71.6±2.2	**76.1±0.2**
Walker2d-medium/50K-medium/0.4M	75.1±1.8	70.3±3.9	**80.1±0.3**
Walker2d-medium/50K-medium/0.6M	74.0±0.6	**79.8±3.6**	**79.1±1.4**
Walker2d-expert/50K-random/0.1M	17.7±12.2	1.8±0.7	**39.5±10.0**
Walker2d-random/50K-expert/0.1M	0.6±0.1	95.1±1.6	**101.4±3.2**
Hopper-medium/50K-medium/0.1M	59.3±2.1	65.6±3.0	**73.9±8.4**
Hopper-medium/50K-medium/0.4M	57.2±1.6	69.2±3.8	**77.8±7.4**
Hopper-medium/50K-medium/0.6M	56.7±1.7	68.6±1.8	**75.9±2.4**
Hopper-expert/50K-random/0.1M	**52.5±4.1**	27.8±14.7	42.7±9.8
Hopper-random/50K-expert/0.1M	4.0±0.5	84.8±10.5	**92.3±9.8**
maze-medium-play(3 tasks) / directed	15.6±1.4	26.8±3.1	**40.0±3.6**
maze-medium-play (3 tasks) / undirected	19.4±2.0	**28.7±3.7**	**29.2±4.1**
maze-medium-diverse(3 tasks) / directed	9.1±3.2	20.0±4.0	**53.2±3.6**
maze-medium-diverse (3 tasks) / undirected	9.9±1.0	**41.9±4.6**	**42.5±5.2**
meta-world / door-open	20.3±10.8	**29.0±13.4**	25.5±15.5
meta-world / door-close	0.0±0.0	109.0±15.6	**114.3±1.8**
meta-world / drawer-open	38.0±65.8	90.0±37.3	**153.8±0.4**
meta-world / drawer-close	**182.3±0.4**	**182.3±0.4**	**182.8±0.4**

Table 5: Comparison between baselines and PDS with the automatic k-adjustment mechanism. Here we use the universal parameter $a = 25$ in all domains. The experimental results are the normalized score metric averaged with five random seeds.

H HOW BAD IS THE REWARD PREDICTION BASELINE ACTUALLY?

We conduct ablation studies for the reward prediction method from three aspects, including model size, ensemble number, and early stopping, which are shown in Table 6, Table 7 and Table 8. For the model size, 256*3 denotes the 256 hidden neurons with three hidden layers. As for the early stopping, Epoch Number = 3 denotes traversing the entire training dataset 3 times. (We find the Epoch Number=3 is enough to reduce the prediction error to a small range, and increasing epochs leads to overfitting.)

We conduct experiments in a setting where the quality of the reward labeled and reward-free data differs significantly. For example, walker2d-expert(50K)-random(0.1M) denotes 50K reward-labeled data from expert datasets and 0.1M reward-free data from random datasets. We set the PDS as the default parameter in all comparisons, including the model size 256*2, the training epoch 3, and the ensemble number 10. All experimental results adopt the normalized score metric averaged over five seeds with standard deviation.

The experimental results show that fine-tuning reward prediction can improve its performance in some cases, Nevertheless, a well-tuned reward prediction method still performs poorly compared to PDS in general. We hypothesize that this is because the "test" (reward-free) dataset may have a large distributional shift from the "training" (reward-labeled) dataset, violating the i.i.d. assumption in supervised learning.

Model Size (Reward Prediction)	256*3	512*3	256*4	512*4	PDS
walker2d-expert(50K)-random(0.1M)	1.8±0.3	11.2±2.0	1.2±0.2	13.3±2.7	**77.6±8.1**
walker2d-random(50K)-expert(0.1M)	95.1±1.9	96.5±1.6	99.4±2.0	97.2±2.2	**105.1±1.2**
hopper-expert(50K)-random(0.1M)	27.8±14.7	31.6±15.2	36.6±14.8	36.3±11.8	**61.5±6.2**
hopper-random(50K)-expert(0.1M)	72.9±19.7	78.9±16.7	**92.1±15.7**	**92.3±14.6**	**93.8±4.9**

Table 6: Ablation for the model size. We adopt various model sizes for the prediction network in the Reward Prediction baseline, while PDS adopts the default parameter model size 256*3.

Epoch Number (Reward Prediction)	1	2	3	PDS
walker2d-expert(50K)-random(0.1M)	2.9±1.3	39.6±4.1	1.8±0.7	**77.6±8.1**
walker2d-random(50K)-expert(0.1M)	91.2±1.3	101.2±1.9	95.1±1.6	**105.1±1.2**
hopper-expert(50K)-random(0.1M)	35.9±9.8	31.0±5.2	27.8±14.7	**61.5±6.2**
hopper-random(50K)-expert(0.1M)	42.4±11.8	57.8±12.2	84.8±10.5	**93.8±4.9**

Table 7: Ablation for the early stopping. We adopt various Training Epochs for the prediction network in the Reward Prediction baseline, while PDS adopts the default parameter epoch number 3.

Environment	Reward Prediction (Ensemble=10)	PDS
walker2d-expert(50K)-random(0.1M)	18.9±3.6	**77.6±8.1**
walker2d-random(50K)-expert(0.1M)	91.41±2.5	**105.1±2.1**
hopper-expert(50K)-random(0.1M)	39.4±5.7	**61.5±6.2**
hopper-random(50K)-expert(0.1M)	78.4±4.6	**93.8±4.9**

Table 8: Ablation for the ensemble. PDS adopts the default parameter ensemble number 10 and we adopt the same ensemble number for the prediction network in the Reward Prediction baseline.