Leveraging Unlabeled Data Sharing through Kernel Function Approximation in Offline Reinforcement Learning

Anonymous authors
Paper under double-blind review

Abstract

Offline reinforcement learning (RL) learns policies from a fixed dataset, but often requires large amounts of data. The challenge arises when labeled datasets are expensive, especially when rewards have to be provided by human labelers for large datasets. In contrast, unlabelled data tends to be less expensive. This situation highlights the importance of finding effective ways to use unlabelled data in offline RL, especially when labelled data is limited or expensive to obtain. In this paper, we present the algorithm to utilize the unlabeled data in the offline RL method with kernel function approximation and give the theoretical guarantee. We present various eigenvalue decay conditions of $\mathcal{H}_k$ which determine the complexity of the algorithm. In summary, our work provides a promising approach for exploiting the advantages offered by unlabeled data in offline RL, whilst maintaining theoretical assurances.

1 Introduction

Reinforcement learning (RL) algorithms have demonstrated empirical success in a variety of domains, including the defeat of Go champions (Silver et al., 2016), robot control (Kalashnikov et al., 2018), and the development of large language models such as ChatGPT (Stiennon et al., 2020). In particular, these achievements are largely associated with online reinforcement learning, characterized by dynamic data collection. However, the widespread adoption of online RL faces significant challenges. In many scenarios, active exploration is impractical due to factors such as the high cost of data collection (Levine et al., 2020). In many scenarios, active exploration is impractical due to factors such as the high cost of data collection (Levine et al., 2020). To this end, in this paper we explore offline reinforcement learning - a fully data-driven framework similar to supervised learning. Unfortunately, fully data-driven offline RL demands large datasets. In more realistic scenarios, offline reinforcement learning (RL) could allow us to use a smaller amount of task-specific data along with a significant amount of task-agnostic data. This data is not labeled with task rewards, and some of it may not be directly relevant to the task at hand.

Prior works use learned classifiers that discriminate between successes and failures for reward labeling (Fu et al., 2018; Singh et al., 2019) in the online RL setting. However, these approaches are unsuitable for the offline RL setting since they require real-time interaction. Alternatively, some research focuses on learning from data without explicit reward labels by directly imitating expert trajectories (Ho & Ermon, 2016; Kostrikov et al., 2019) or deriving the reward function through inverse reinforcement learning using an expert dataset (Fu et al., 2017; Finn et al., 2016). However, in real-world scenarios, these approaches may face challenges due to the resource-intensive and costly nature of the expert trajectory acquisition and reward labelling process.

Yu et al. (2022) has revealed the challenges associated with learning to predict rewards, highlighting the surprising efficacy of setting the reward to zero. Despite these findings, the impact of reward prediction methods on performance and the potential demonstrable benefits of reward-free data in offline reinforcement learning (RL) remain unclear. In response to this, Hu et al. (2023) have introduced a novel model-free approach named Provable Data Sharing (PDS). PDS incorporates uncertainty penalties into the learned reward functions, maintaining a conservative algorithm. This method allows PDS to take advantage of

unlabeled data for offline RL, especially in linear MDPs. However, the linear MDP assumption is inflexible and rarely is fulfilled in practice. This question naturally arises.

How can we enhance the performance of offline RL algorithms that use kernel function approximation by effectively using reward-free data?

This work focuses on the episodic Markov decision process (MDP). The reward function and value function are both represented by kernel functions. Inspired by the Provable Data Sharing (PDS) (Hu et al., 2023) framework, we propose a new algorithm. The PDS algorithm has two main components. First, it pessimistically estimates rewards by applying additional penalties to the reward function learned from labeled data. This augmentation is designed to prevent overestimation, thus ensuring a conservative algorithm. The second part of the PDS algorithm uses the Pessimistic Value Iteration (PEVI) algorithm introduced by Jin et al. (2021) to derive the policy. Our main contribution is that

- Extension of PDS framework: We expand the applicability of the Provable Data Sharing (PDS) framework, initially introduced by Hu et al. (2023). This extension goes beyond the original linear Markov Decision Process (MDP) setting, incorporating kernel function approximation. This expansion enhances the versatility of the PDS framework, making it applicable to a broader range of scenarios. Our derivation is influenced by methodologies proposed for kernelized contextual bandits (Chowdhury & Gopalan, 2017; Valko et al., 2013; Srinivas et al., 2009), as well as techniques such as pessimistic value iteration (PEVI) (Jin et al., 2021) and the kernel optimum least squares value iteration algorithm (KOVI) (Yang et al., 2020).
- Focus on finite-horizon MDPs: While Hu et al. (2023) concentrates on a discounted infinite-horizon MDP setting, our work shifts the focus to finite-horizon MDPs. This adjustment accommodates horizon-dependent reward functions and transition probability functions, addressing a specific and practical aspect of reinforcement learning.
- Feature coverage assessment via concentratability coefficient: In contrast to Hu et al. (2023), which assumes the concentratability coefficient to be bounded for assessing coverage over the state action space, we measure the distribution shift using the spectrum of feature covariance matrices. This alternative metric (Wang et al., 2020a), as in Assumption 4.8, is well-established in supervised learning and particularly suitable for scenarios involving linear function approximation.
- Data-splitting technique: We introduce a data-splitting technique to mitigate potential challenges associated with the logarithmic covering number in the learning bound, as discussed by Xie et al. (2021).

Our research provides a theoretical guarantee for effectively utilizing the benefits of reward-free data in offline RL. We aim to enhance the robustness of offline RL methods by maintaining theoretical guarantees, which offers a valuable contribution to the ongoing development of more resilient and efficient RL frameworks.

2 Related Works

The issue of suboptimality in discounted and episodic MDP with a model has been considered in linear and kernel settings. The results are presented in Table 1. In the episodic MDP setting, we have the dataset with N trajectories of horizon H, and the suboptimality dependent on N and H. On the other hand, in a discounted MDP setting, we have the dataset with length N, and suboptimality dependent on N. The PEVI algorithm (Jin et al., 2021) serves as the foundational algorithm within Hu et al. (2023) and our work. If we assume that the infinite horizon MDP should conclude within H steps (referred to as the effective horizon) (Yan et al., 2022), we can set the discount factor γ such that $H = 1/(1-\gamma)$. Consequently, the suboptimality for the PDS algorithm is expressed as $\tilde{\mathcal{O}}(dH^2 N_2^{-\frac{1}{2}})$ where N_2 is the number of trajectories for the unlabeled dataset. Similar to Hu et al. (2023), we incorporate unsupervised data sharing to enhance the offline RL algorithm. The linear setting is a special case of the kernel setting with a linear kernel. In this case, we can recover the suboptimality as $\tilde{\mathcal{O}}(Hd^{\frac{1}{2}} N_1^{-\frac{1}{2}})$, where N_1 is the number of trajectories for the labeled dataset, as

provided in Hu et al. (2023). A notable difference between PEVI and PDS lies in PDS's utilization of data sharing to improve the suboptimality through an unlabeled dataset. It's important to note that $N_2 > N_1$ in general. When comparing PDS with our approach in a linear setting, the H-folds data splitting in our algorithm enhances the suboptimality by a factor of $\sqrt{d}$. However, this improvement comes with a tradeoff, as our algorithm introduces a suboptimality increment by a factor of $\sqrt{H}$ because we need to partition the data set into H folds. As a result, each estimated value function is derived from only N_2/H episodes of data.

Algorithm	MDP	Setting	SubOpt
PEVI (Jin et al., 2021)	Episodic	Linear	$\tilde{\mathcal{O}}(dH^2 N_1^{-\frac{1}{2}})$
PDS (Hu et al., 2023)	Discounted	Linear	$\tilde{\mathcal{O}}(d^{\frac{1}{2}}(1-\gamma)^{-1}N_1^{-\frac{1}{2}}) + \tilde{\mathcal{O}}(d(1-\gamma)^{-2}N_2^{-\frac{1}{2}})$
Our work	Episodic	kernel-based, d-finite spectrum	$\tilde{\mathcal{O}}(Hd^{\frac{1}{2}}N_1^{-\frac{1}{2}}) + \tilde{\mathcal{O}}(H^{\frac{5}{2}}d^{\frac{1}{2}}N_2^{-\frac{1}{2}})$
Our work	Episodic	kernel-based, general setting	$\tilde{\mathcal{O}}(H\sqrt{G(N_1,\nu)\zeta_{\mathcal{D}_1}}) + \tilde{\mathcal{O}}(H^2\sqrt{G(\frac{N}{H},\lambda)\zeta_{\widetilde{\mathcal{D}}}})$

Table 1: The existing suboptimality under weak convergence (see Assumption 4.8)(except for the last row) , discussed in Section 2. Here, the labeled dataset represented as $\{(s'^{\tau}_h, a'^{\tau}_h, r^{\tau}_h)\}^{N_1,H}_{\tau,h=1}$, unlabeled dataset represented as $\{(s'^{\tau+N_1}_h, a'^{\tau+N_1}_h)\}^{N_2,H}_{\tau,h=1}$, and $\mathcal{D}^{\theta}$, which is a combination of labeled dataset and unlabeled dataset with $N = N_1 + N_2$ trajectories. We partition the dataset $\mathcal{D}^{\theta}$ into H disjoint and equally sized sub dataset $\{\widetilde{\mathcal{D}}^{\theta}_h\}^{H}_{h=1}$. Denote γ as the discount factor for discounted MDP, $G(N,\lambda)$ is the maximum information gain, $\zeta_{\mathcal{D}} = \max_{h\in[H]} \zeta_h(\mathcal{D}', \mathcal{D})$ represents a maximum amount of information from the dataset $\mathcal{D}$ and $\mathcal{D}'$, where $\mathcal{D}'$ is the combination of $\mathcal{D}$ and observed data z, and $\zeta_{\widetilde{\mathcal{D}}} = \max_{h\in[H]} \zeta_h((\widetilde{\mathcal{D}}^{\theta}_h)', \widetilde{\mathcal{D}}^{\theta}_h)$. Note that $\nu = 1 + \frac{1}{N_1}$ and $\lambda = 1 + \frac{1}{N}$. In a linear MDP setting, it is stated that the transition probability can be represented linearly in a feature map of state-action with d dimensions.

Offline Reinforcement Learning

In offline reinforcement learning (RL), the goal is to learn a policy from a static data set collected previously without interacting with the environment. Current approaches in offline RL (Levine et al., 2020) can be broadly classified into dynamic programming methods and model-based methods. Dynamic programming methods aim to learn a state action value function, known as the Q function. Subsequently, this value function is used either to directly find the optimal policy or, in the case of actor-critic methods, to estimate a gradient for the expected returns of a policy. The offline dynamic programming algorithm operates in a tabular setting (Jin et al., 2018). However, algorithms designed for tabular settings have limitations when applied to function approximation settings with a large number of effective states. Recent work has centered around the functional approximation setting, especially in the linear setting, where the value function (or transition model) can be represented using a linear function of a known feature mapping (Jin et al., 2021; Cai et al., 2020; Zanette et al., 2021). As the linear Markov decision process (MDP) assumption is rigid and rather restrictive in practice, Wang et al. (2020b) explores the kernel optimal least squares value iteration (KOVI) algorithm (Yang et al., 2020) for general function approximation. In contrast, model-based methods rely on their ability to estimate the transition function using a parameterized model, such as a neural network. Instead of employing dynamic programming methods to fit the model, model-based approaches leverage their ability to effectively utilize large and diverse datasets to estimate the transition function (Yu et al., 2021b; Janner et al., 2019). Both of the methods presented above require a large amount of data to learn a state-action or transition function. In our work, we use reward-free data (i.e., unlabeled data) to improve the performance of learning a state-action function.

Offline Data Sharing

Data sharing strategies in multi-task reinforcement learning (RL) have shown effectiveness, as observed in works such as Yu et al. (2021a); Eysenbach et al. (2020); Chen et al. (2021). This involves reusing data across different tasks by relabeling rewards, thereby enhancing performance in multi-task offline RL scenarios. Prior

work has employed various relabeling strategies. These include uniform labeling (Kalashnikov et al., 2021), labeling based on metrics such as estimated Q-values (Yu et al., 2021a), and labeling based on distances to states in goal-conditioned settings (Chen et al., 2021). However, these approaches either necessitate access to the functional form of the reward for relabeling or are confined to goal-conditioned settings. On the other hand, Yu et al. (2022) proposes a straightforward strategy by assigning zero rewards to unlabeled data. On the other hand, Hu et al. (2023) employs linear regression to label rewards for unlabeled data. These approaches present alternative and potentially simpler methods for relabeling, especially in scenarios where direct access to the reward function is challenging or unavailable. In our work, we propose kernel ridge regression to exploit unlabeled data which under certain conditions can be reduced to linear regression.

3 Background

3.1 Episodic Markov Decision Process

Consider an episodic MDP (Yang et al., 2020; Sutton & Barto, 2018), denoted as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, \mathcal{P}, r)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, horizon H, transition function $\mathcal{P} = \{\mathcal{P}_h\}_{h\in[H]}$, and reward function $r = \{r_h\}_{h\in[H]}$. We assume that the reward function is bounded, that is, $r_h \in [0,1]$. For any policy $\pi = \{\pi_h\}_{h\in[H]}$ and $h \in [H]$, we define the state-value function $V_h^\pi : \mathcal{S} \to \mathbb{R}$ and the action-valued function (Q-function) $Q_h^\pi : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$ as $V_h^\pi(s) = \mathbb{E}_\pi\left[\sum_{t=h}^H r_t(s_t,a_t)|s_h = s\right]$ and $Q_h^\pi(s,a) = \mathbb{E}_\pi\left[\sum_{t=h}^H r_t(s_t,a_t)|s_h = s, a_h = a\right]$. These two functions satisfy the well-known Bellman equation: $V_h^\pi(s) = \langle Q_h^\pi(s,\cdot), \pi_h(\cdot \mid s)\rangle_{\mathcal{A}}$ and $Q_h^\pi(s,a) = \mathbb{E}\left[r_h(s_h,a_h) + V_{h+1}^\pi(s_{h+1}) \mid s_h = s, a_h = a\right]$. For any function $f : \mathcal{S} \to \mathbb{R}$, we define the transition operator at each step $h \in [H]$ as $(\mathbb{P}_h f)(s,a) = \mathbb{E}\left[f(s_{h+1}) \mid s_h = s, a_h = a\right]$, and define the Bellman operator as $(\mathbb{B}_h f)(s,a) = \mathbb{E}\left[r_h(s_h,a_h) \mid s_h = s, a_h = a\right] + (\mathbb{P}_h f)(s,a)$. Similarly, for all $h \in [H]$, the Bellman optimality equations defined as $V_h^*(s) = \sup_{a\in\mathcal{A}} Q_h^*(s,a)$ and $Q_h^*(s,a) = \left(\mathbb{B}_h V_{h+1}^*\right)(s,a)$. Meanwhile, the optimal policy π^* satisfies $\pi_h^*(\cdot \mid s) = \underset{\pi_h}{\operatorname{argmax}} \langle Q_h^*(s,\cdot), \pi_h(\cdot \mid s)\rangle_{\mathcal{A}}$ and $V_h^*(s) = \langle Q_h^*(s,\cdot), \pi_h^*(\cdot \mid s)\rangle_{\mathcal{A}}$. Reinforcement learning aims to learn a policy maximizing expected cumulative reward. Accordingly, we define the performance metric(i.e.,suboptimality) as

$$\mathrm{SubOpt}(\pi; s) = V_1^{\pi^*}(s) - V_1^\pi(s). \tag{1}$$

3.2 Assumption of Offline Data

In offline RL setting, a learner uses pre-collected dataset $\mathcal{D}$, which consists of N trajectories $\{(s_h^\tau, a_h^\tau, r_h^\tau)\}_{\tau,h=1}^{N,H}$, generated by some fixed but unknown MDP $\mathcal{M}$ under the behavior policy π^{b} in the following manner: $s_1^\tau \sim \rho^{\mathrm{b}}, a_h^\tau \sim \pi_h^{\mathrm{b}}(\cdot \mid s_h^\tau)$ and $s_{h+1}^\tau \sim \mathcal{P}_h(\cdot \mid s_h^\tau, a_h^\tau), 1 \le h \le H$. Here ρ^{b} represents a predetermined initial state distribution associated with the static dataset. The learner may also have partial observations of the reward in addition to the above state-action observations. More elaborately, we assume access to both a labeled dataset $\mathcal{D}_1 = \left\{\left(s'^\tau_h, a'^\tau_h, r_h^\tau\right)\right\}_{\tau,h=1}^{N_1,H}$, and an unlabeled dataset $\mathcal{D}_2 = \left\{\left(s'^{\tau+N_1}_h, a'^{\tau+N_1}_h\right)\right\}_{\tau,h=1}^{N_2,H}$. We utilize the estimated reward function with parameter θ, as determined in section 4, to relabel dataset $\mathcal{D}_2$. The relabeled dataset, denoted as $\mathcal{D}_2^\theta = \left\{\left(s'^{\tau+N_1}_h, a'^{\tau+N_1}_h, \widehat{r}_h^{\theta_h}(s'^{\tau+N_1}_h, a'^{\tau+N_1}_h)\right)\right\}_{\tau,h=1}^{N_2,H}$.

3.3 Reproducing Kernel Hilbert Space

Consider a reproducing kernel Hilbert space (RKHS) as a function space. For simplicity, let $z = (s,a)$ denote a state-action pair and denote $\mathcal{Z} = \mathcal{S}\times\mathcal{A}$. Without loss of generality, we regard $\mathcal{Z}$ as a compact subset of $\mathbb{R}^m$, where the dimension m is fixed. Let $k : \mathcal{Z}\times\mathcal{Z} \to \mathbb{R}$ be a positive definite continuous kernel and its corresponding kernel matrix $[\mathbf{K}]_{i,j} = k(z_i, z_j), \forall i,j \in [m]$. Note that $\mathbf{K}$ is positive semi-definite. Define $\mathcal{H}_k$ as the RKHS induced by k, containing a family of functions defined in $\mathcal{Z}$. Let $\langle\cdot,\cdot\rangle_{\mathcal{H}_k} : \mathcal{H}_k \times \mathcal{H}_k \to \mathbb{R}$ and $\|\cdot\|_{\mathcal{H}_k} : \mathcal{H}_k \to \mathbb{R}$ denote the inner product and the norm on $\mathcal{H}_k$, respectively. According to the reproducing property, for all $f \in \mathcal{H}_k$, and $z \in \mathcal{Z}$, holds $f(z) = \langle f, k(\cdot, z)\rangle_{\mathcal{H}_k}$. For more details and different characterizations of RKHS, see Aronszajn (1950); Berlinet & Thomas-Agnan (2011). Without loss of generality, we assume that $\sup_{z\in\mathcal{Z}} k(z,z) \le 1$.

Let $\mathcal{L}^2(\mathcal{Z})$ be the set of square-integrable functions on $\mathcal{Z}$ with respect to the Lebesgue measure and let $\langle , \rangle_{\mathcal{L}^2}$ be the inner product on $\mathcal{L}^2(\mathcal{Z})$. The kernel function k induces an integral operator $T_k : \mathcal{L}^2(\mathcal{Z}) \to \mathcal{L}^2(\mathcal{Z})$ defined as $T_k f(z) = \int_{\mathcal{Z}} k(z, z') f(z') \, \mathrm{d}z'$ for all $f \in \mathcal{L}^2(\mathcal{Z})$. By Mercer's theorem (Steinwart & Christmann, 2008), the integral operator T_k has countable and positive eigenvalues $\{\sigma_i\}_{i \geq 1}$ and the corresponding eigenfunctions $\{\psi_i\}_{i \geq 1}$. Then, the kernel function admits a spectral expansion $k(z, z') = \sum_{i=1}^{\infty} \sigma_i \psi_i(z) \psi_j(z')$. Moreover, the RKHS $\mathcal{H}_k$ can be written as a subset of $\mathcal{L}^2(\mathcal{Z})$ such that $\mathcal{H}_k = \left\{ f \in \mathcal{L}^2(\mathcal{Z}) : \sum_{i=1}^{\infty} \frac{\langle f, \psi_i \rangle_{\mathcal{L}^2}^2}{\sigma_i} < \infty \right\}$, and the inner product of $\mathcal{H}_k$ also can be written as $\langle f, g \rangle_{\mathcal{H}_k} = \sum_{i=1}^{\infty} (1/\sigma_i) \langle f, \psi_i \rangle_{\mathcal{L}^2} \langle g, \psi_i \rangle_{\mathcal{L}^2}$ for all $f, g \in \mathcal{H}_k$. With the above construction, the scaled eigenfunctions $\{\sqrt{\sigma_i}\psi_i\}_{i \geq 1}$ form an orthonormal basis for $\mathcal{H}_k$. We define the mapping $\phi : z \mapsto k(z, \cdot)$ to transform data from $\mathcal{Z} = \mathcal{S} \times \mathcal{A}$ to the (possibly infinite-dimensional) RKHS $\mathcal{H}_k$, which satisfies $k(z, z') = \langle \phi(z), \phi(z') \rangle_{\mathcal{H}_k}$ for all $z, z' \in \mathcal{Z}$ (Steinwart & Christmann, 2008, Lemma 4.19). We define the maximum information gain (Srinivas et al., 2009) to describe the complexity of $\mathcal{H}_k$:

$$G(n, \lambda) = \sup \left\{ \frac{1}{2} \log \det \left(I + K_{\mathcal{D}} / \lambda \right) : \mathcal{D} \subset \mathcal{Z}, |\mathcal{D}| \leq n \right\}, \tag{2}$$

where $K_{\mathcal{D}}$ is the kernel matrix for the set $\mathcal{D}$. Furthermore, the magnitude of maximal information gain $G(n, \lambda)$ depends on how rapidly the eigenvalues decay to zero, serving as a proxy dimension of $\mathcal{H}$ in the case of an infinite-dimensional space. If $\mathcal{H}_k$ is of finite rank, we have that $G(n, \lambda) = \mathcal{O}(d \log n)$ (Yang et al., 2020), where d is the rank of $\mathcal{H}_k$ – referred as the d-finite spectrum. In the following, we present several conditions that are often used in the analysis of the RKHS property of $\mathcal{H}_k$ (Yang et al., 2020; Vakili et al., 2021; Yeh et al., 2023) characterizing the eigenvalue decay of $\mathcal{H}_k$.

Assumption 3.1. *The integral operator T_K has eigenvalues $\{\sigma_j\}_{j \geq 1}$ and the associated eigenfunctions $\{\psi_j\}_{j \geq 1}$. We assume that $\{\sigma_j\}_{j \geq 1}$ satisfies one of the following conditions for some constant $d > 0$.*

- *d-finite spectrum: $\sigma_j = 0, \forall j > d$, where d is a positive integer.*
- *d-exponential decay: there exists some constants $C_1, C_2 > 0$ such that $\sigma_j \leq C_1 \cdot \exp\left(-C_2 \cdot j^d\right)$, $\forall j \geq 1$, where $d > 0$.*
- *d-polynomial decay: there exists some constants $C_1 > 0$ such that $\sigma_j \leq C_1 \cdot j^{-d}$ $\forall j \geq 1$,, where $d > 1$.*

For both d-exponential decay and d-polynomial decay, we assume that there exists $C_\psi > 0$ such that $\sup_{z \in \mathcal{Z}} \sigma_j^\tau \cdot |\psi_j(z)| \leq C_\psi$ holds for all $j \geq 1$ and $\tau \in [0, 1/2)$.

3.4 Pessimistic Value Iteration and Kernel Setting

We consider the pessimistic value iteration, i.e., PEVI (Jin et al., 2021) algorithm, described in Algorithm 2, as the backbone algorithm. This is a model-free, theoretically guaranteed offline algorithm. The fundamental insight of PEVI lies in the incorporation of a penalty function, which essentially introduces a sense of pessimism, into the value iteration algorithm. The key challenge to extend PEVI to kernel setting is that the dimension (even effective dimension) of the kernel based model (when interpreted as linear model) is divergent. In addition, we apply the data splitting method (Rashidinejad et al., 2021; Xie et al., 2021). As introduced in Rashidinejad et al. (2021), data splitting makes sure that the estimated value $\widehat{V}_{h+1}$ and estimated Bellman operator $\widehat{\mathbb{B}}_h$ are estimated using different subsets of $\mathcal{D}$, this yields conditional independence that is required in bounding concentration terms of the form $\left(\widehat{\mathbb{B}}_h - \mathbb{B}_h\right) \widehat{V}_{h+1}$, and hence the suboptimality can be reduced by a factor of $\sqrt{d}$. However, applied naively, this data splitting induces one undesired $\sqrt{H}$ factor in the optimality as we need to split $\mathcal{D}$ into H folds and thus each $\mathbb{B}_h$ is estimated using only N/H episodes of data. Further details of the PEVI algorithm can be found in Appendix B.

4 Unsupervised Data Sharing

Our algorithm comprises two main components. The first part involves employing kernel ridge regression to learn the reward function using the labeled dataset and constructing the confidence set. Next, to mitigate

overestimation in reward prediction, we construct the pessimism reward parameter $\tilde{\theta}$ within the confidence set. Section 4.1 discusses this in more detail. The second part involves using the pessimistic reward estimator $\tilde{\theta}$ to relabel the entire dataset, which is a combination of the labeled dataset and the relabeled dataset. Following this, we employ the PEVI algorithm with kernel approximation and data splitting (refer to Algorithm 3) to determine the optimal policy. The detailed steps of the algorithm are outlined in Algorithm 1.

Algorithm 1 Data Sharing, Kernel Approximation

1: **Data:** Labeled dataset $\mathcal{D}_1$, and unlabeled dataset $\mathcal{D}_2$.
2: **Input:** Parameter $\beta_h(\delta), \delta, B, \nu, \lambda$.
3: Define $\mathcal{D}^\theta$, which is a combination of the labeled dataset $\mathcal{D}_1$ and the unlabeled dataset $\mathcal{D}_2^\theta$, and partition the dataset $\mathcal{D}^\theta$ into H disjoint and equally sized sub datasets $\{\widetilde{\mathcal{D}}_h^\theta\}_{h=1}^H$.
4: Learn the reward function $\widehat{\theta}_1, \cdots, \widehat{\theta}_H$ from $\mathcal{D}_1$ with

$$\widehat{\theta}_h = \underset{\theta_h \in \mathcal{H}_k}{\operatorname{argmin}} \sum_{\tau=1}^{N_1} \left[r_h^\tau - \widehat{r}_h^{\theta_h}(s'^\tau_h, a'^\tau_h) \right]^2 + \nu \|\theta_h\|_{\mathcal{H}_k}^2. \tag{3}$$

5: Construct the pessimistic reward function with parameter $\tilde{\theta} := \{\tilde{\theta}_h\}_{h=1}^H$ satisfy

$$\tilde{r}_h^{\tilde{\theta}_h}(s,a) = \max\left\{ \left\langle \widehat{\theta}_h, \phi(s,a) \right\rangle_{\mathcal{H}_k} - \beta_h(\delta) \left\| (\Lambda_h^{\mathcal{D}_1})^{-\frac{1}{2}} \phi(s,a) \right\|_{\mathcal{H}_k}, 0 \right\}. \tag{4}$$

6: Annotate the reward in $\mathcal{D}^\theta$ with parameter $\theta = \widetilde{\theta}$.
7: Learn the policy from the relabeled dataset $\mathcal{D}^{\widetilde{\theta}}$ using Algorithm 3 in Appendix.

$$\{\widehat{\pi}_h\}_{h=1}^H \leftarrow \mathrm{PEVI}\left(\mathcal{D}^{\widetilde{\theta}}, B, \lambda\right). \tag{5}$$

8: **Result:** $\widehat{\pi} = \{\widehat{\pi}_h\}_{h=1}^H$.

4.1 Pessimistic Reward Estimation

We utilize labeled dataset $\mathcal{D}_1$ to train a reward function $\widehat{r}_h^{\theta_h}$, using it to label the unlabeled data. Assume that the observed reward is generated as $r_h^\tau = r_h(s'^\tau_h, a'^\tau_h) + \epsilon_h^\tau$ where $r_h : (s,a) \mapsto \langle \theta_h^*, \phi(s,a) \rangle_{\mathcal{H}_k}$ satisfies $r_h(s,a) \in [0,1]$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, and ϵ_h^τ are i.i.d. centered 1-SubGaussian noise. Here $\theta_h^* \in \mathcal{H}_k$ is an unknown parameter, and $\phi : \mathcal{S} \times \mathcal{A} \to \mathcal{H}_k$ is a known feature map defined in Section 3.3. Furthermore, we assume that $\|\theta_h^*\|_{\mathcal{H}_k} \leq \mathcal{S}$. We learn the reward function from labeled data through a kernel ridge regression problem. Using the feature representation, we write

$$\widehat{\theta}_h = \underset{\theta_h \in \mathcal{H}_k}{\operatorname{argmin}} \sum_{\tau=1}^{N} \left[r_h^\tau - \widehat{r}_h^{\theta_h}(s'^\tau_h, a'^\tau_h) \right]^2 + \nu \|\theta_h\|_{\mathcal{H}_k}^2, \tag{6}$$

where $\widehat{r}_h^{\theta_h}(s,a) = \langle \phi(s,a), \theta_h \rangle_{\mathcal{H}_k}$ with parameter θ_h. However, this method leads to an overestimation of predicted reward values, as highlighted in Yu et al. (2022). A novel algorithm called Provable Data Sharing (PDS) is introduced in Hu et al. (2023) to mitigate this problem. PDS incorporates uncertainty penalties into the learned reward functions and integrates seamlessly with existing offline RL algorithms in a linear MDP setting. We extend the application of this algorithm to the kernel setting.

To address the problem of overestimating predicted rewards, we analyze the uncertainty in the learned reward function. The previous solution defines the center of the ellipsoidal confidence set:

$$\mathcal{C}_h(\delta) = \left\{ \theta \in \mathcal{H}_k : \left\| \theta - \hat{\theta}_h \right\|_{\Lambda_h^{\mathcal{D}_1}} \leq \beta_h(\delta) \right\}, \tag{7}$$

where $\|\theta\|^2_{\Lambda_h^{\mathcal{D}_1}} = \left\langle \theta, \Lambda_h^{\mathcal{D}_1}\theta \right\rangle_{\mathcal{H}_k}$ and $\Lambda_h^{\mathcal{D}_1} = \sum_{\tau=1}^{N_1} \phi(s'^{\tau}_h, a'^{\tau}_h)\phi(s'^{\tau}_h, a'^{\tau}_h)^\top + \nu I_{\mathcal{H}_k}$ is a positive definite operator, and $\beta_h(\delta)$ is its radius which follows Proposition 4.1.

Proposition 4.1. *We define $\beta_h(\delta)$ with the labeled data set $\mathcal{D}_1$ by $\beta_h(\delta) = \sqrt{\nu}\mathcal{S} + \sqrt{\log \frac{\det\left[\nu I + K_h^{\mathcal{D}_1}\right]}{\delta^2}}$, where $K_h^{\mathcal{D}_1}$ is the Gram matrix constructed from the dataset $\mathcal{D}_1$ as $\left[K_h^{\mathcal{D}_1}\right]_{\tau,\tau'} = k(z'^{\tau}_h, z'^{\tau'}_h)$, where $z'^{\tau}_h = (s'^{\tau}_h, a'^{\tau}_h)$ for $\tau, \tau' \in [N_1]$ and for each $h \in [H]$ and $\delta \in (0,1)$. Then, with probability at least $1-\delta$ we have $\left\|\widehat{\theta}_h - \theta^*_h\right\|_{\Lambda_h^{\mathcal{D}_1}} \leq \beta_h(\delta)$, where $\widehat{\theta}_h$ is the solution of Equation (6). Furthermore, consider the information gain $G(N, \nu)$, defined in Equation (2) of the matrix $K_h^{\mathcal{D}_1}$ and set $\nu = 1 + 1/N_1$, $\beta_h(\delta)$ is rewritten as*

$$\sqrt{\nu}\mathcal{S} + \sqrt{2G(N_1, \nu) + 1 + \log\frac{1}{\delta^2}}. \tag{8}$$

Moreover, define $\mathcal{C}_h(\delta) = \left\{\theta \in \mathcal{H}_k : \left\|\theta - \widehat{\theta}_h\right\|_{\Lambda_h^{\mathcal{D}_1}} \leq \beta_h(\delta)\right\}$, we have $\mathbb{P}(\theta^*_h \in C_h(\delta)) \geq 1-\delta$.

Proof. Please refer to Appendix B.1 for detailed proof. □

In Proposition 4.1, the uncertainty of the learned reward function depends on the maximum information gain of the kernel matrix $K_h^{\mathcal{D}_1}$. However, finding the optimal parameter within the confidence set is computationally inefficient. To address this challenge, Hu et al. (2023) proposes an approach that preserves the pessimistic property of the offline algorithm. This method uses pessimistic estimation, allowing the algorithm to remain pessimistic while mitigating computational challenges. Formally, we construct the pessimistic reward function $\tilde{r}_h^{\tilde{\theta}_h}(s,a)$ for the parameter $\tilde{\theta}_h$ as

$$\tilde{r}_h^{\tilde{\theta}_h}(s,a) = \max\left\{\left\langle \widehat{\theta}_h, \phi(s,a)\right\rangle_{\mathcal{H}_k} - \beta_h(\delta)\left\|(\Lambda_h^{\mathcal{D}_1})^{-\frac{1}{2}}\phi(s,a)\right\|_{\mathcal{H}_k}, 0\right\}. \tag{9}$$

The equation (9) is guaranteed by the following lemma derived from Cauchy-Schwarz inequalities.

Lemma 4.2. $\left|\left\langle \theta_h - \widehat{\theta}_h, \phi(s,a)\right\rangle_{\mathcal{H}_k}\right| \leq \beta_h(\delta)\left\|(\Lambda_h^{\mathcal{D}_1})^{-\frac{1}{2}}\phi(s,a)\right\|_{\mathcal{H}_k}$ *for any* $\theta_h \in \mathcal{C}_h(\delta), h \in [H]$.

The equation (9) provides a lower bound for the reward function within the confidence set $\mathcal{C}(\delta)$. When the labeled data is scarce, or when there is a significant shift in the distribution between the labeled and unlabeled data, the confidence interval becomes wider and then the equation (9) degenerates to 0, which is reduced to the UDS algorithm (Yu et al., 2022; Hu et al., 2023).

4.2 Theoretical Analysis

We assume that the Bellman operator maps any bounded function onto a bounded RKHS norm ball, which is the common assumption using in the function approximation (Yang et al., 2020; Jin et al., 2018).

Assumption 4.3. *Define the function class $\mathcal{Q}^* = \{f \in \mathcal{H}_k : \|f\|_{\mathcal{H}_k} \leq R_Q H\}$ for some fixed constant $R_Q > 0$. Then, for any $h \in [H]$ and any $Q : \mathcal{S}\times\mathcal{A} \to [0,H]$, it holds that $\mathbb{B}_h V \in \mathcal{Q}^*$ for $V(s) = \max_{a\in\mathcal{A}} Q(s,a)$.*

A sufficient condition for Assumption 4.3 to hold is when $\mathcal{S} = [0,1]^m$ and that $r_h(\cdot,\cdot), \mathcal{P}_h(s' \mid \cdot,\cdot) \in \{f \in \mathcal{H}_k : \|f\|_{\mathcal{H}_k} \leq 1\}$ for all $h \in [H], \forall s' \in \mathcal{S}$. To see this, suppose this condition holds, then for any integrable $V : \mathcal{S} \to [0,H]$ holds,

$$\begin{aligned}
\|r_h + \mathbb{P}_h V\|_{\mathcal{H}_k} &\leq \|r_h\|_{\mathcal{H}_k} + \|\mathbb{P}_h V\|_{\mathcal{H}_k} \leq 1 + \left\|\int_{s'\in\mathcal{S}} \mathcal{P}_h(s'|\cdot,\cdot)V(s')\, ds'\right\|_{\mathcal{H}_k} \\
&\leq 1 + \int_{s'\in\mathcal{S}} \|\mathcal{P}_h(s'|\cdot,\cdot)V(s')\|_{\mathcal{H}_k}\, ds' = 1 + \int_{s'\in\mathcal{S}} \|\mathcal{P}_h(s'|\cdot,\cdot)\|_{\mathcal{H}_k}\|V(s')\|_{\mathcal{H}_k}\, ds' \\
&\leq 1 + H\int_{s'\in\mathcal{S}} ds' = H + 1.
\end{aligned}$$

Note that under the assumptions of measurability and boundedness on the kernel k, $\|\mathbb{P}_h V\|_{\mathcal{H}_k} \in \mathcal{H}_k$, which is given in Muandet et al. (2017, section 3.1). Thus, Assumption 4.3 holds with $R_Q = 2$. This assumption is mild and is also used in Yang et al. (2020). Similar assumptions are used in linear MDP's, which are much stricter (Jin et al., 2021; Zanette et al., 2020). The suboptimality of the Algorithm 1 is characterized by the following theorem.

Theorem 4.4. *Consider the MDP described in Section 3.1. Under Assumption 3.1 and Assumption 4.3, and suppose the labeled dataset $\mathcal{D}_1$ and unlabeled dataset $\mathcal{D}_2^\theta$ are defined in Section 3.2. Define $\mathcal{D}^\theta = \{(s_h^\tau, a_h^\tau, \widehat{r}_h^{\theta_h}(s_h^\tau, a_h^\tau))\}_{\tau,h=1}^{N,H}$, which is a combination of labeled dataset $\mathcal{D}_1$ and unlabeled dataset $\mathcal{D}_2^\theta$ with $N = N_1 + N_2$. We partition dataset $\mathcal{D}^\theta$ into H disjoint and equally sized sub dataset $\{\widetilde{\mathcal{D}}_h^\theta\}_{h=1}^H$, where $|\widetilde{\mathcal{D}}_h^\theta| = N_h = N/H$. Let $\mathcal{I}_h = \{N_h \cdot (h-1) + 1, \ldots, N_h \cdot h\} = \{\tau_{h,1}, \cdots, \tau_{h,N_h}\}$ satisfy $\widetilde{\mathcal{D}}_h^\theta = \{(s_h^\tau, a_h^\tau, r_h^\tau)\}_{\tau \in \mathcal{I}_h}$. We set $\lambda = 1 + \frac{1}{N}, \nu = 1 + \frac{1}{N_1}$ in Algorithm 1, where*

$$\beta_h(\delta) = \begin{cases} \sqrt{1+\frac{1}{N_1}}\mathcal{S} + \sqrt{C_1 \cdot d \cdot \log N_1 + \log(\frac{1}{\delta^2})} & \text{d-finite spectrum,} \\ \sqrt{1+\frac{1}{N_1}}\mathcal{S} + \sqrt{C_1 \cdot (\log N_1)^{1+\frac{1}{d}} + \log(\frac{1}{\delta^2})} & \text{d-exponential decay,} \\ \sqrt{1+\frac{1}{N_1}}\mathcal{S} + \sqrt{C_1 \cdot (N_1)^{\frac{m+1}{d+m}} \cdot \log(N_1) + \log(\frac{1}{\delta^2})} & \text{d-polynomial decay.} \end{cases} \tag{10}$$

$$B = \begin{cases} C_2 \cdot H \cdot \sqrt{d \log(N/\delta)} & \text{d-finite spectrum,} \\ C_2 \cdot H \cdot \sqrt{(\log N/\delta)^{1+1/d}} & \text{d-exponential decay,} \\ C_2 \cdot N^{\frac{m+1}{2(d+m)}} H^{1-\frac{m+1}{2(d+m)}} \cdot \sqrt{\log(N/\delta)} & \text{d-polynomial decay.} \end{cases} \tag{11}$$

Here, $C_1, C_2 > 0$ are absolute constants that does not depend on N_1, N, nor H. Then, for fixed initial state $s_0 \in \mathcal{S}$, with probability $1 - 2\delta$, the policy $\hat{\pi}$ generated by Algorithm 1 satisfies

$$\begin{aligned} \mathrm{SubOpt}(\hat{\pi}; s_0) \leq\ & 2\sum_{h=1}^H \beta_h(\delta) \mathbb{E}_{\pi^*}\left[\|\phi(s_h, a_h)\|_{(\Lambda_h^{\mathcal{D}_1})^{-1}} \mid s_1 = s_0\right] \\ & + 2B\sum_{h=1}^H \mathbb{E}_{\pi^*}\left[\|\phi(s_h, a_h)\|_{(\Lambda_h^{\widetilde{\mathcal{D}}_h^\theta})^{-1}} \mid s_1 = s_0\right], \end{aligned} \tag{12}$$

Proof. For a detailed proof, see Appendix B.2. □

Two key terms express the suboptimality bound. The first term is the reward bias introduced by uncertainties in estimating rewards. This term reflects the challenges and inaccuracies associated with predicting or estimating rewards in a given environment. The second term represents the offline algorithm and optimal policy π^* error.

Remark 4.5. *We use the Lemma C.2 to rewrite the term of $\beta_h(\delta)$ and B in the Theorem 4.4 as $\beta_h(\delta) = \tilde{\mathcal{O}}(\sqrt{G(N_1, 1+\frac{1}{N_1})})$ and $B = \tilde{\mathcal{O}}(H\sqrt{G(N, 1+\frac{1}{N})})$.*

By Remark 4.5, both terms $\beta_h(\delta)$ and B depend on the kernel function class. It is worth noting that the term $\|\phi(s_h, a_h)\|_{(\Lambda_h^{\mathcal{D}})^{-1}}$ can be expressed as an information quantity for the dataset $\mathcal{D}$, as outlined in Lemma 4.6.

Proposition 4.6. *For all $h \in [H]$, we partition dataset $\mathcal{D}$ into H disjoint and equally sized sub datasets $\{\widetilde{\mathcal{D}}_h\}_{h=1}^H$, where $\widetilde{\mathcal{D}}_h = \{(s_h^\tau, a_h^\tau, r_h^\tau)\}_{\tau \in \mathcal{I}_h}$ with $\mathcal{I}_h = \{N_h \cdot (h-1) + 1, \ldots, N_h \cdot h\} = \{\tau_{h,1}, \cdots, \tau_{h,N_h}\}$ and $N_h = N/H$. Denote the operator $\Phi_h^{\widetilde{\mathcal{D}}_h} : \mathcal{H}_k \to \mathbb{R}^{N_h}$, and $\Lambda_h^{\widetilde{\mathcal{D}}_h} : \mathcal{H}_k \to \mathcal{H}_k$ as*

$$\Phi_h^{\widetilde{\mathcal{D}}_h} = \begin{pmatrix} \phi\left(z_h^{\tau_{h,1}}\right)^\top \\ \vdots \\ \phi\left(z_h^{\tau_{h,N_h}}\right)^\top \end{pmatrix} = \begin{pmatrix} k\left(\cdot, z_h^{\tau_{h,1}}\right)^\top \\ \vdots \\ k\left(\cdot, z_h^{\tau_{h,N_h}}\right)^\top \end{pmatrix}, \ \Lambda_h^{\widetilde{\mathcal{D}}_h} = \lambda \cdot I_{\mathcal{H}} + (\Phi_h^{\widetilde{\mathcal{D}}_h})^\top \Phi_h^{\widetilde{\mathcal{D}}_h}. \tag{13}$$

Define gram matrix $K_h^{\widetilde{\mathcal{D}}_h} = \Phi_h^{\widetilde{\mathcal{D}}_h}(\Phi_h^{\widetilde{\mathcal{D}}_h})^\top$. *Then, for any* $z \in \mathcal{Z}$, *we have*

$$\phi(z)^\top(\Lambda_h^{\widetilde{\mathcal{D}}_h})^{-1}\phi(z) \leq 2 \cdot \left[\log\det\left(I + K_h^{\widetilde{\mathcal{D}}'_h}/\lambda\right) - \log\det\left(I + K_h^{\widetilde{\mathcal{D}}_h}/\lambda\right)\right], \tag{14}$$

where $\widetilde{\mathcal{D}}'_h$ *is the combination of dataset* $\widetilde{\mathcal{D}}_h$ *and* z *which satisfies* $\Lambda_h^{\widetilde{\mathcal{D}}'_h} = \Lambda_h^{\widetilde{\mathcal{D}}_h} + \phi(z)\phi(z)^\top$.

Proof. For a detailed proof, see Appendix B.3. □

Remark 4.7. *In Proposition 4.6,* $\mathcal{H}_k$ *can be infinite dimensional. However, for the sake of clarity, we represent* $\Phi_h^{\widetilde{\mathcal{D}}_h}$ *as a matrix and* $\phi(z_h^\tau)$ *as a column vector for all* $\tau \in \mathcal{I}_h$.

Here, we define

$$\zeta_h(\mathcal{D}', \mathcal{D}) = 2\left[\log\det\left(I + K_h^{\mathcal{D}'}/\lambda\right) - \log\det\left(I + K_h^{\mathcal{D}}/\lambda\right)\right], \tag{15}$$

as the maximal information amount between the dataset $\mathcal{D}'$ and $\mathcal{D}$. Proposition 4.6 states that if the training data set is well known about z, then Equation (15) will be close to zero. On the other hand, if the training data set is not well known about z, then Equation (15) will be large.

We specialize the d-finite spectrum case of Theorem 4.4 under a weak data coverage assumption to better understand the convergence of Algorithm 1.

Assumption 4.8 (Weak Convergence)**.** *Suppose the dataset* $\mathcal{D} = \{(s_h^\tau, a_h^\tau, r_h^\tau)\}_{\tau,h=1}^{N,H}$ *consists of* N *trajectories, for all* $h \in [H]$, *the trajectories are drawn independently and identically from distributions induced by some fixed behavior policy* $\bar{\pi}$ *such that there exists a constant* $c_{min} > 0$ *satisfying* $\inf_{\|f\|_{\mathcal{H}_k}=1}\langle f, \mathbb{E}_{\bar{\pi}}\left[\phi(z_h)\phi(z_h)^\top\right] f\rangle \geq c_{min}$ *for any* $h \in [H]$.

Intuitively, Assumption 4.8 posits that the collected data should be relatively well distributed throughout the state action space. Notably, assumption 4.8 shares similarities with other explorability assumptions common in reinforcement learning literature, such as those in Yin et al. (2022); Wagenmaker & Pacchiano (2023).

Corollary 4.9 (Well-Explored Dataset)**.** *In the d-finite spectrum case, assume that the Assumption 4.8 holds under the same conditions as Theorem 4.4. Then for* $N_1 \geq \Omega(\log(dH/\delta))$ *and* $N \geq H \cdot \Omega(\log(dH/\delta))$, *with probability at least* $1-\delta$, *we have*

$$\begin{aligned}\mathrm{SubOpt}(\widehat{\pi}; s) &\leq 2\beta_h(\delta) \cdot H \cdot c'/\sqrt{N_1} + 2B \cdot H \cdot c'/\sqrt{N_h} \\ &\leq \tilde{\mathcal{O}}(H\sqrt{\frac{d}{N_1}}) + \tilde{\mathcal{O}}(H^{\frac{5}{2}}\sqrt{\frac{d}{N_2}}).\end{aligned} \tag{16}$$

In the d-finite spectrum case, a significant difference between our present study and previous work (Hu et al., 2023) lies in the incorporation of factors $\sqrt{d}$ and $\sqrt{H}$, introduced by the implementation of the data splitting technique (Xie et al., 2021). This technique plays a crucial role in the linear case, influencing the overall convergence behavior of the learned policy. If we aim to transform the feature mapping from a dimensionality of d to d', where $d' > d$. In this context, the data partitioning method can help mitigate the convergence of the error bound. Finally, we combine the result in Therorem 4.4, Remark 4.5, and Corollary 4.9 to get the Table 1.

Notice that in the case of d-exponential and d-polynomial decay, if Assumption 4.8 holds true, by integrating Lemma B.2, Lemma B.3, and equation 93, we can deduce the explicit form of $\mathrm{SubOpt}(\widehat{\pi}; s)$ as $\tilde{\mathcal{O}}\left(HN_1^{-\frac{1}{2}}\right) + \tilde{\mathcal{O}}\left(H^{\frac{5}{2}}N_2^{-\frac{1}{2}}\right)$ and $\tilde{\mathcal{O}}\left(HN_1^{-\frac{1}{2}+\frac{m+1}{d+m}}\right) + \tilde{\mathcal{O}}\left(H^{\frac{5}{2}-\frac{m+1}{2(d+m)}}N_2^{-\frac{1}{2}+\frac{m+1}{2(d+m)}}\right)$, correspondingly. Nonetheless, Assumption 4.8 does not hold under scrutiny. To demonstrate this, let's assume that Assumption 4.8 is true. It means that for every f within the set $\{\|f\|_{\mathcal{H}_k} = 1\}$, it satisfies $\mathbb{E}_{\tilde{\pi}}[\langle f, \phi(z_h)\phi(z_h)^\top f\rangle] \geq c_{\min}$. Then, we express ϕ and f as $\phi = \sum_{i=1}^\infty a_i\psi_i$ and $f = \sum_{i=1}^\infty b_i\psi_i$ respectively, where $\{\psi_i\}_{i=1}^\infty$ is orthonormal basis of $\mathcal{H}_k$. Given that f can represent any function satisfying $\|f\|_{\mathcal{H}_k} = 1$, let f be any vector such that $f = b_j\psi_j$ for an arbitrary j. Consequently, for all j, the expectation $\mathbb{E}_{\tilde{\pi}}[\langle f, \phi(z_h)\phi(z_h)^\top f\rangle] = a_j^2 \geq c_{\min}$ is satisfied, which results in a paradox because the norm should be finite; however, $\|\phi\|_{\mathcal{H}_k} = \sum_{j=1}^\infty {a_j}^2 \geq \sum_{j=1}^\infty c_{min} = \infty$.

5 Conclusion

In this paper, we demonstrate that incorporating unlabeled data into offline RL can greatly improve offline RL performance. Our theoretical analysis shows how unlabeled data can improve the performance of offline RL, especially in a more general function approximation setting, in contrast to the results in Hu et al. (2023). Our analysis is based on the common offline RL assumption about the dataset, providing a comprehensive examination of the algorithm's performance under these conditions. In future work, it may be interesting to extend to the discounted MDP setting to deal with more category problems and the low-rank MDP (Uehara et al., 2021).

References

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

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68 (3):337–404, 1950.

Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

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

Annie S Chen, Suraj Nair, and Chelsea Finn. Learning generalizable robotic reward functions from" in-the-wild" human videos. *Robotics: Science and Systems (RSS)*, 2021.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pp. 844–853. PMLR, 2017.

Yaqi Duan, Zeyu Jia, and Mengdi Wang. Minimax-optimal off-policy evaluation with linear function approximation. In *International Conference on Machine Learning*, pp. 2701–2709. PMLR, 2020.

Ben Eysenbach, Xinyang Geng, Sergey Levine, and Russ R Salakhutdinov. Rewriting history with inverse rl: Hindsight inference for policy improvement. *Advances in neural information processing systems*, 33: 14783–14795, 2020.

Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International conference on machine learning*, pp. 49–58. PMLR, 2016.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. In *International Conference on Learning Representations*, 2017.

Justin Fu, Avi Singh, Dibya Ghosh, Larry Yang, and Sergey Levine. Variational inverse control with events: A general framework for data-driven reward definition. *Advances in neural information processing systems*, 31, 2018.

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. The provable benefits of unsupervised data sharing for offline reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=MTTPLcwvqTt`.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? *Advances in neural information processing systems*, 31, 2018.

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

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.

Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. Imitation learning via off-policy distribution matching. In *International Conference on Learning Representations*, 2019.

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.

Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, Bernhard Schölkopf, et al. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2): 1–141, 2017.

Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, 34:11702–11716, 2021.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Avi Singh, Larry Yang, Kristian Hartikainen, Chelsea Finn, and Sergey Levine. End-to-end robotic reinforcement learning without reward engineering. *environment (eg, by placing additional sensors)*, 2019.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, 2009.

Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

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

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Joel A Tropp et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

Masatoshi Uehara, Xuezhou Zhang, and Wen Sun. Representation learning for online and offline rl in low-rank mdps. In *International Conference on Learning Representations*, 2021.

Sattar Vakili, Kia Khezeli, and Victor Picheny. On information gain and regret bounds in gaussian process bandits. In *International Conference on Artificial Intelligence and Statistics*, pp. 82–90. PMLR, 2021.

Michal Valko, Nathaniel Korda, Rémi Munos, Ilias Flaounas, and Nelo Cristianini. Finite-time analysis of kernelised contextual bandits. In *Uncertainty in Artificial Intelligence*, 2013.

Andrew Wagenmaker and Aldo Pacchiano. Leveraging offline data in online reinforcement learning. In *International Conference on Machine Learning*, pp. 35300–35338. PMLR, 2023.

Ruosong Wang, Dean P Foster, and Sham M Kakade. What are the statistical limits of offline rl with linear function approximation? In *International Conference on Learning Representations*, 2020a.

Ruosong Wang, Ruslan Salakhutdinov, and Lin F Yang. Provably efficient reinforcement learning with general value function approximation. *arXiv preprint arXiv:2005.10804*, 2020b.

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

Yuling Yan, Gen Li, Yuxin Chen, and Jianqing Fan. Model-based reinforcement learning is minimax-optimal for offline zero-sum markov games. *arXiv preprint arXiv:2206.04044*, 2022.

Zhuoran Yang, Chi Jin, Zhaoran Wang, Mengdi Wang, and Michael I Jordan. On function approximation in reinforcement learning: Optimism in the face of large state spaces. *arXiv preprint arXiv:2011.04622*, 2020.

Sing-Yuan Yeh, Fu-Chieh Chang, Chang-Wei Yueh, Pei-Yuan Wu, Alberto Bernacchia, and Sattar Vakili. Sample complexity of kernel-based q-learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 453–469. PMLR, 2023.

Ming Yin, Yaqi Duan, Mengdi Wang, and Yu-Xiang Wang. Near-optimal offline reinforcement learning with linear representation: Leveraging variance information with pessimism. In *International Conference on Learning Representations*, 2022.

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. *Advances in neural information processing systems*, 34:28954–28967, 2021b.

Tianhe Yu, Aviral Kumar, Yevgen Chebotar, Karol Hausman, Chelsea Finn, and Sergey Levine. How to leverage unlabeled data in offline reinforcement learning. In *International Conference on Machine Learning*, pp. 25611–25635. PMLR, 2022.

Andrea Zanette, David Brandfonbrener, Emma Brunskill, Matteo Pirotta, and Alessandro Lazaric. Frequentist regret bounds for randomized least-squares value iteration. In *International Conference on Artificial Intelligence and Statistics*, pp. 1954–1964. PMLR, 2020.

Andrea Zanette, Ching-An Cheng, and Alekh Agarwal. Cautiously optimistic policy optimization and exploration with linear function approximation. In *Conference on Learning Theory*, pp. 4473–4525. PMLR, 2021.