Sample Complexity of Kernel-Based Q-Learning

Sing-Yuan Yeh[1,2] **Fu-Chieh Chang**[1,3] **Chang-Wei Yueh**[4] **Pei-Yuan Wu**[3,4] **Alberto Bernacchia**[1] **Sattar Vakili**[1]

[1]MediaTek Research

[2]Graduate Program of Data Science, National Taiwan University and Academia Sinica
[3]Graduate Institute of Communication Engineering, National Taiwan University
[4]Department of Electrical Engineering, National Taiwan University

Abstract

Modern reinforcement learning (RL) often faces an enormous state-action space. Existing analytical results are typically for settings with a small number of state-actions, or simple models such as linearly modeled Q-functions. To derive statistically efficient RL policies handling large state-action spaces, with more general Q-functions, some recent works have considered nonlinear function approximation using kernel ridge regression. In this work, we derive sample complexities for kernel based Q-learning when a generative model exists. We propose a nonparametric Q-learning algorithm which finds an ϵ-optimal policy in an arbitrarily large scale discounted MDP. The sample complexity of the proposed algorithm is order optimal with respect to ϵ and the complexity of the kernel (in terms of its information gain). To the best of our knowledge, this is the first result showing a finite sample complexity under such a general model.

1 INTRODUCTION

In recent years, Reinforcement Learning (RL) has been successfully applied to several fields, including gaming (Silver et al., 2016; Lee et al., 2018; Vinyals et al., 2019), autonomous driving (Kahn et al., 2017), microchip design (Mirhoseini et al., 2021), robot control (Kalashnikov et al., 2018), and algorithm search (Fawzi et al., 2022). Real-world problems usually contain an enormous state-action space, possibly infinite. For example, the game of *Go* has 10^{170} states (Silver et al., 2016), and the number of actions in the space of algorithms for matrix multiplication is 10^{30} (Fawzi et al., 2022). It is currently not fully understood how RL algorithms are able to learn successful policies to solve these problems. Modern function approximators, such as kernel-based learning and deep neural networks, seem to be required for this success.

An important theoretical question is as follows. Consider a Markov decision process (MDP) with an unknown transition probability distribution. Suppose that a generative model (Kakade, 2003) is available, which provides sample transitions from any state-action pair. How many samples are required to learn a sufficiently good policy? That is referred to as the sample complexity.

Previous works have derived theoretical bounds on the sample complexity, under certain simple settings such as tabular and linear MDPs. In the tabular setting, it was shown that the sample complexity of learning an ϵ-optimal policy (that is the value function is at most ϵ away from the optimal value function) is in $\mathcal{O}(\frac{|\mathcal{S}||\mathcal{A}|}{\epsilon^2})$, where $|\mathcal{S}|$ and $|\mathcal{A}|$ are the cardinality of the state and action sets, respectively (Kearns and Singh, 1998; Azar et al., 2013; Sidford et al., 2018a,b; Agarwal et al., 2020), implying that for a very large state-action space, a virtually infinite number of samples is required to obtain a good performance. Another line of work considers a linear MDP model, where the transition probability admits a linear representation in a d-dimensional state-action feature map. It was shown that the sample complexity is in $\mathcal{O}(\frac{d}{\epsilon^2})$ in this case, that is independent of the size of state and action spaces (Yang and Wang, 2019). Unfortunately, the linear assumption is rather inflexible and not often the case in practice.

In order to address the limitations of small state-actions or simple models, arising from tabular and linear MDP assumptions, a few recent studies considered nonlinear function approximation over possibly infinite state-action domains using kernel ridge regression. In these works, the transition probability distribution (and sometimes the reward function) are flexibly represented using a kernel-

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

based model (Yang and Wang, 2020; Yang et al., 2020a,b). The kernel-based models provide powerful regressor and uncertainty estimates, which can be leveraged to guide the RL algorithm. Furthermore, kernel-based models have a great representation capacity and can model a wide range of problems, considering that all continuous functions on compact subsets of $\mathbb{R}^d$ can be approximated using common kernels (Srinivas et al., 2010). The existing works, however, do not address the specific question of sample complexity considered in this work, and instead derive regret bounds in the setting of an episodic MDP. A more detailed comparison is provided in Section 1.2.

The kernel-based approaches may be understood as a linear model with an infinite-dimensional state-action feature map, that corresponds to, e.g., the Mercer eigenfeatures of the kernel (see Section 2.2). In this sense, the linear model is a special case of the kernel-based model with a linear kernel. Nonetheless, the results on the sample complexity of linear MDPs do not extend to the kernel-based models, as those sample complexities scale with the dimension of the feature map (that is possibly infinite in the kernel setting). In contrast, in the kernel setting, the sample complexity depends on certain kernel-specific properties determined by the complexity of the kernel, which will be discussed in more detail.

1.1 Contributions

Considering a discounted MDP and the question of sample complexity (similar to Azar et al., 2013; Sidford et al., 2018a,b; Yang and Wang, 2019), we extend and generalize the existing work as follows.

- We introduce Kernel-based Q-Learning, referred to as KQLearn, a sample collection algorithm, which returns an ϵ-optimal policy with a finite sample complexity over a very general class of models. In comparison to tabular and linear MDP settings, KQLearn makes at least two innovative contributions. In the tabular setting, the samples are collected from all state-action pairs that leads to an $|\mathcal{S}||\mathcal{A}|$ scaling of the sample complexity. In the linear setting, the samples are collected from a set of state-actions spanning the entire state-action space (leading to the scaling of the sample complexity with dimension d). Then, an estimation of the parameters of the linear model are updated through value iteration. Neither of these approaches are feasible in our case with an infinite state-action space and a non-parametric kernel-based model. KQLearn instead takes advantage of uncertainties provided by the kernel model to create a finite state-action set which is used for collecting the samples. These samples are then passed through an approximate Bellman operator using kernel ridge regression to update the value function (that is a continuous function over the entire state-action space).
- We derive a finite sample complexity for KQLearn under a wide range of kernel models. In particular, we consider two classes of kernels with exponentially ($\sigma_m \sim \exp(-m^{\beta_e})$, $\beta_e > 0$) and polynomially ($\sigma_m \sim m^{-\beta_p}$, $\beta_p > 1$) decaying Mercer eigenvalues σ_m (see Definition 2). We prove a sample complexity of $\tilde{\mathcal{O}}(\frac{1}{\epsilon^2})$ and $\tilde{\mathcal{O}}\left((\frac{1}{\epsilon})^{\frac{2\beta_p}{\beta_p-1}}\right)$[1] under these two settings, respectively. To the best of our knowledge, this is the first finite sample complexity, for all $\beta_p > 1$, and the first order optimal sample complexity in ϵ, under the setting of polynomially decaying eigenvalues. Comparison with the related work is discussed in more detail in Section 1.2. As a special case, we recover the $\tilde{\mathcal{O}}(\frac{d}{\epsilon^2})$ sample complexity of the linear setting reported in Yang and Wang (2019).

We acknowledge that our bounds on the sample complexity of KQLearn may not be order optimal in the dependence on the discount factor γ. In particular, our bounds grow with $\frac{1}{(1-\gamma)^7}$ in the case of smooth kernels, similar to the PPQ-Learning algorithm proposed in Yang and Wang (2019) for the linear setting. Under the tabular and linear settings, however, this dependency was improved to $\frac{1}{(1-\gamma)^3}$, in Sidford et al. (2018a) and Yang and Wang (2019), respectively. It appears a challenging problem whether the same improvement is feasible here. As mentioned above, even establishing a finite sample complexity is a challenging problem, and the sample complexities in the existing work may diverge with difficult kernels (some polynomial kernels as discussed in Section 1.2).

1.2 Related Work

The specific problem of sample complexity in a discounted MDP using a generative model has been considered in tabular and linear settings. The results are summarized in Table 1. Other variants of the problem, consider MDPs in the absence of a generative model (e.g., see, Azar et al., 2017; Jin et al., 2018, 2020; Russo, 2019; Yang et al., 2020a,b; Kakade et al., 2020; Zhou et al., 2021; Domingues et al., 2021, as representative works, as well as references therein), often episodic, with T episodes of length H, and regret bounds depending on T and H. The regret bounds can then be translated into sample complexities (e.g., see, Jin et al., 2018; Yang et al., 2020b). These results are also reported in Table 1. Other approaches to nonlinear function approximation in RL include models with bounded *eluder* dimension (Wang et al., 2020; Ayoub et al., 2020) and smoothing kernels (Domingues et al., 2021). Among these works the two most relevant ones to ours are Yang and Wang (2019) and Yang et al. (2020a,b).

[1] The notations $\mathcal{O}$ and $\tilde{\mathcal{O}}$ are used to denote the mathematical order, and that up to hiding logarithmic factors, respectively.

Table 1: The existing sample complexities under various settings, discussed in Section 1.2.

ALGORITHM	MDP	SETTING	SAMPLE COMPLEXITY
(Q-learning with UCB, Jin et al., 2018)	Episodic	Tabular	$\tilde{\mathcal{O}}\left(\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert H^4}{\epsilon^2}\right)$
(LSVI-UCB, Jin et al., 2020)	Episodic	Linear	$\tilde{\mathcal{O}}\left(\frac{d^3 H^4}{\epsilon^2}\right)$
(KOVI, Yang et al., 2020b)	Episodic	Kernel-based, polynomial eigendecay	$\tilde{\mathcal{O}}\left(H^4(\frac{1}{\epsilon})^{\frac{2\beta_p}{\beta_p-2}}\right)$
		Kernel-based, exponential eigendecay	$\tilde{\mathcal{O}}\left(\frac{H^4}{\epsilon^2}\right)$
(Sidford et al., 2018a, Variance-Reduced QVI)	Discounted	Tabular	$\tilde{\mathcal{O}}\left(\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^3\epsilon^2}\right)$
(PPQ-Learning, Yang and Wang, 2019)	Discounted	Linear	$\tilde{\mathcal{O}}\left(\frac{d}{(1-\gamma)^7\epsilon^2}\right)$
(OPPQ-Learning, Yang and Wang, 2019)	Discounted	Linear	$\tilde{\mathcal{O}}\left(\frac{d}{(1-\gamma)^3\epsilon^2}\right)$
KQLearn	Discounted	Kernel-based, polynomial eigendecay	$\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^{\frac{2\beta_p}{\beta_p-1}}(1-\gamma)^{\frac{7\beta_p-1}{\beta_p-1}}}\right)$
		Kernel-based, exponential eigendecay	$\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^2(1-\gamma)^7}\right)$

Similar to Yang and Wang (2019), we also consider sample complexity in a discounted MDP using a generative model. We consider a non-parametric kernel-based model, while they considered a parametric linear model. Thus, neither their algorithm nor their results extend to our setting. The linear setting is a special case of the kernel setting with a linear kernel, in which, we recover the $\tilde{\mathcal{O}}(\frac{d}{\epsilon^2})$ sample complexity, given in Yang and Wang (2019), for two algorithms: PPQ-Learning and OPPQ-Learning. The latter improved the sample complexity with respect to the discount factor.

Similar to Yang et al. (2020b), we also consider a kernel-based model. We consider sample complexity in a discounted MDP with a generative model, while they primarily considered regret bounds in an episodic MDP. They also reported sample complexities as a direct consequence of their regret bounds. Specifically, under the two settings of exponentially and polynomially decaying eigenvalues, their sample complexities translate to $\tilde{\mathcal{O}}(\frac{1}{\epsilon^2})$ and $\tilde{\mathcal{O}}\left((\frac{1}{\epsilon})^{\frac{2\beta_p}{\beta_p-2}}\right)$, respectively. Under the polynomial setting, their sample complexity bound is larger than ours. In addition, their sample complexity is not always finite and may diverge when $1 < \beta_p \leq 2$, that includes many cases of interest. This suboptimality and possibly trivial result is a consequence of the superlinear (thus, trivial) regret bounds when $1 < \beta_p \leq 2$. See Vakili et al. (2021d), for a detailed discussion on the theoretical challenges related to this result.

For example, consider the Matérn family of kernels as one of the most commonly used (Snoek et al., 2012; Shahriari et al., 2015) and theoretically interesting (Srinivas et al., 2010) family of kernels. For a Matérn kernel with smoothness parameter ν on a d dimensional input domain, $\beta_p = 1 + \frac{2\nu}{d}$ (Yang et al., 2020b). That implies the sample complexity in Yang et al. (2020b) diverges when $d > 2\nu$ (that is often the case when using the Matérn kernel). We, however, emphasize that the discounted MDP with a generative model and the episodic MDPs are different settings, and cannot be compared directly. Nonetheless, we present the first always finite sample complexity under a very general setting covering all kernels with polynomially decaying eigenvalues.

Another related problem is the kernel-based bandit problem (Srinivas et al., 2010), which corresponds to a degenerate MDP with $|\mathcal{S}| = 1$. The kernel-based bandit problem is a well studied problem with order optimal regret bounds (Salgia et al., 2021; Li and Scarlett, 2022) and sample complexities (Vakili et al., 2021a). The lower bounds on sample complexities for the squared exponential (SE) and Matérn kernels are reported in (Scarlett et al., 2017), which have the same scaling with ϵ (up to logarithmic factors) as in our results, showing the order optimality of our sample complexities with ϵ (see Section 4). To obtain the optimal order of sample complexities, Vakili et al. (2021a) proposed the Maximum Variance Reduction (MVR) algorithm, which selects observation points with the highest uncertainty in a greedy manner. We build upon their approach by extending it to the sample complexities of Q-learning. This extension is significantly more involved due to the presence of Markovian dynamics and the Bellman operator.

There exist other studies in the field of kernel-based RL that are related to our work, including Ormoneit and Sen (2002); Domingues et al. (2021). However, there are notable differences between their work and ours in terms of both the settings and the algorithms. Specifically, these works assumed a Lipschitz reward function and a Lipschitz transition probability distribution, which allowed them to use kernel smoothing methods. In contrast, our work uses kernel ridge regression with the assumption of a bounded RKHS norm of the transition probability distribution.

Paper structure: In Section 2, the problem is formalized, after an overview of the background on MDPs and kernel ridge regression. In Section 3, KQLearn is presented. The results are discussed in Section 4. A high level analysis is provided in Section 5, while the details are deferred to the appendix.

2 PRELIMINARIES

In this section, we overview the background on MDPs and kernel ridge regression. We then formally state the problem of sample complexity for Q-learning under this setting.

2.1 Discounted Markov Decision Process

A discounted Markov Decision Process (MDP) can be described by the tuple $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\gamma \in (0, 1)$ is the discount factor, $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the reward function and $P(\cdot|s, a)$ is the transition probability distribution[2] on $\mathcal{S}$ for the next state from state-action pair (s, a). We use the notation $\mathcal{Z} = \mathcal{S} \times \mathcal{A}$ to denote the state-action space. Our results generally hold true for (possibly very large and) finite $\mathcal{Z}$ or certain infinite $\mathcal{Z}$. For correctness, we assume that $\mathcal{Z}$ is a compact subset of $\mathbb{R}^d$.

The goal is to find a (possibly random) policy $\pi : \mathcal{S} \to \mathcal{A}$, that maximizes the long-term expected reward, i.e., the value function,

$$V^\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, \pi\left(s_t\right)\right) \mid s_0 = s\right],$$

where $s_t \sim P(\cdot|s_{t-1}, \pi(s_{t-1}))$ forms the trajectory of the states. It can be shown that (e.g., see Puterman, 2014), under mild assumptions (e.g., continuity of P, compactness of $\mathcal{Z}$, and boundedness of r) there exists an optimal policy π^* which attains the maximal possible value V^* at every state,

$$\forall s \in \mathcal{S}: \quad V^*(s) := V^{\pi^*}(s) = \max_{\pi} V^\pi(s).$$

[2]We intentionally do note use the standard term *transition kernel* for P, to avoid confusion with the term *kernel* in kernel-based learning.

To simplify the notation, for a value function $V : \mathcal{S} \to \mathbb{R}$, let

$$[PV](s, a) := \mathbb{E}_{s' \sim P(\cdot|s,a)}[V(s')].$$

The Q-function, also sometimes referred to as the state-action value function, of a policy π, and the optimal Q-function are defined as

$$Q^\pi(s, a) = r(s, a) + \gamma[PV^\pi](s, a), \text{ and}$$
$$Q^*(s, a) = Q^{\pi^*}(s, a),$$

respectively. The Bellman operator $\mathcal{T} : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{S}}$ is defined as

$$\forall s \in \mathcal{S}: \quad [\mathcal{T}V](s) = \max_{a \in \mathcal{A}} \left\{r(s, a) + \gamma[PV](s, a)\right\}.$$

Sample complexity of ϵ-optimal policies: An ϵ-optimal policy is defined as follows.

Definition 1. *(ϵ-optimal policy) A policy π is called ϵ-optimal if it achieves near optimal values from any initial state as follows:*

$$V^\pi(s) \geq V^*(s) - \epsilon, \quad \forall s \in \mathcal{S},$$

or equivalently $\|V^\pi - V^*\|_\infty \leq \epsilon$.

We aim to learn ϵ-optimal policies using a small number of samples. In this work, following Kearns and Singh (1998); Azar et al. (2013); Sidford et al. (2018a,b); Yang and Wang (2019), we suppose that a generative model (Kakade, 2003) is given where the RL algorithm is able to query transition samples $s' \sim P(\cdot|s, a)$ for any state-action pair $(s, a) \in \mathcal{Z}$. The *sample complexity* of an RL algorithm is defined as the number of such samples used by the algorithm to obtain an ϵ-optimal policy.

2.2 RKHS and Kernel Ridge Regression

The existing work achieving finite sample complexity in the RL setting typically assumes a small state-action space or linearly modeled MDPs. These results can be generalized and extended using kernel-based learning. In particular, a natural approach is to use elements of a known reproducing kernel Hilbert space (RKHS) to model the transitions. In this section, we overview RKHSs and kernel ridge regression.

Let $K : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ be a known positive definite kernel with respect to a finite Borel measure. Let $\mathcal{H}_K$ be the RKHS induced by K, where $\mathcal{H}_K$ contains a family of functions defined on $\mathcal{Z}$. 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 of $\mathcal{H}_K$, respectively. The reproducing property implies that for all $f \in \mathcal{H}_K$, and $z \in \mathcal{Z}$, $\langle f, K(\cdot, z)\rangle_{\mathcal{H}_K} = f(z)$. Without loss of generality, we assume $K(z, z) \leq 1$ for all z. Mercer theorem implies, under certain mild conditions, K can be represented using an infinite dimensional feature map:

$$K(z,z') = \sum_{m=1}^{\infty} \sigma_m \psi_m(z)\psi_m(z'). \tag{1}$$

A formal statement and the details are provided in Appendix A.

Kernel ridge regression: Kernel-based models provide powerful regressor and uncertainty estimators (roughly speaking, surrogate posterior variances) which can be leveraged to guide the RL algorithm. In particular, consider an unknown function $f \in \mathcal{H}_K$. Consider a set $\mathcal{U}_J = \{z_j\}_{j=1}^J \subset \mathcal{Z}$ of J inputs. Assume J noisy observations $\{Y(z_j) = f(z_j) + \epsilon_j\}_{j=1}^J$ are provided, where ϵ_j are i.i.d. zero mean sub-Gaussian noise terms. Kernel ridge regression provides the following regressor and uncertainty estimate, respectively (see, e.g., Schölkopf et al., 2002),

$$\begin{aligned}\hat{f}_{\mathcal{U}_J}(z) &= k_{\mathcal{U}_J}^\top(z)(K_{\mathcal{U}_J} + \lambda^2 I_J)^{-1} Y_{\mathcal{U}_J},\\ \Sigma^2_{\mathcal{U}_J}(z) &= K(z,z) - k_{\mathcal{U}_J}^\top(z)(K_{\mathcal{U}_J} + \lambda^2 I_J)^{-1} k_{\mathcal{U}_J}(z),\end{aligned} \tag{2}$$

where $k_{\mathcal{U}_J}(z) = [K(z,z_1),\ldots,K(z,z_J)]^\top$ is a $J \times 1$ vector of the kernel values between z and observations, $K_{\mathcal{U}_J} = [K(z_i,z_j)]_{i,j=1}^J$ is the $J \times J$ kernel matrix, $Y_{\mathcal{U}_J} = [Y(z_1),\ldots,Y(z_J)]^\top$ is the $J \times 1$ observation vector, I_J is the identity matrix of dimensions J, and $\lambda > 0$ is a free regularization parameter.

Confidence intervals: The prediction and uncertainties provided by kernel ridge regression allow us to use standard confidence intervals in the algorithm and analysis. In particular, various results exist stating that with probability at least $1-\delta$, the prediction function satisfies $|f(z) - \hat{f}_{\mathcal{U}_J}(z)| \le \beta(\delta)\Sigma_{\mathcal{U}_J}(z)$ (either for fixed z, or simultaneously for all z) where the confidence interval width multiplier $\beta(\delta)$ depends on the properties of the observation noise and the complexity of f in terms of its RKHS norm (Srinivas et al., 2010; Abbasi-Yadkori, 2013; Vakili et al., 2021a, 2022). If the domain $\mathcal{Z}$ is finite, the uniform confidence bounds readily follow from a union bound over the confidence intervals for a fixed z. For continuous domains, a discretization argument is typically used considering the following continuity assumption.

Assumption 1. *For each $n \in \mathbb{N}$, there exists a discretization $\mathbb{Z}$ of $\mathcal{Z}$ such that, for any $f \in \mathcal{H}_K$ with $\|f\|_{\mathcal{H}_K} \le C_K$, we have $f(z) - f([z]) \le \frac{1}{n}$, where $[z] = \arg\min_{z' \in \mathbb{Z}} ||z' - z||_{l^2}$ is the closest point in $\mathbb{Z}$ to z, and $|\mathbb{Z}| \le cC_K^d n^d$, where c is a constant independent of n and C_K.*

Assumption 1 is a technical and mild assumption that holds for typical kernels such as SE and Matérn with $\nu > 1$ (Srinivas et al., 2010; Chowdhury and Gopalan, 2017; Vakili et al., 2021a).

In our analysis, we use the following confidence interval for the RKHS elements.

Lemma 1 (Vakili et al. (2021a, 2022))**.** *Consider a fixed design of observation points where $\mathcal{U}_J$ is independent of the observation noise. When the noise terms are sub-Gaussian with parameter R [3] and $\|f\|_{\mathcal{H}_K} \le C_K$, the following each hold uniformly in $z \in \mathcal{Z}$, with probability $1-\delta$,*

$$\begin{aligned} f(z) &\le \hat{f}_{\mathcal{U}_J}(z) + \beta(\delta)\Sigma_{\mathcal{U}_J}(z),\\ f(z) &\ge \hat{f}_{\mathcal{U}_J}(z) - \beta(\delta)\Sigma_{\mathcal{U}_J}(z),\end{aligned} \tag{3}$$

where $\beta(\delta) = \mathcal{O}\left(C_K + \frac{R}{\lambda}\sqrt{d\log(\frac{JC_K}{\delta})}\right)$.

Maximal information gain: It is useful for our analysis to define maximal information gain $\Gamma_{K,\lambda}$, that is a kernel specific complexity term. It allows us to bound the total uncertainty in the kernel model using results similar to elliptical potential lemma (Carpentier et al., 2020). In particular, let us define

$$\Gamma_{K,\lambda}(J) = \sup_{\mathcal{U}\subset\mathcal{Z}, |\mathcal{U}|\le J} \frac{1}{2}\log\det\left(I_J + \frac{1}{\lambda^2}K_{\mathcal{U}}\right). \tag{4}$$

Then, we have the following.

Lemma 2 (Srinivas et al. (2010))**.** *For any set $\mathcal{U}_J \subset \mathcal{Z}$, we have*

$$\sum_{j=1}^{J} \Sigma^2_{\mathcal{U}_{j-1}}(z_j) \le \frac{2}{\log(1+1/\lambda^2)}\Gamma_{K,\lambda}(J). \tag{5}$$

2.3 Problem Formulation

Consider the discounted MDP described in Section 2.1. We are interested in designing an algorithm with a small sample complexity which obtains an ϵ-optimal RL policy, under the assumption that the transition probability distribution lives in the RKHS of a known kernel. Without loss of generality, we assume its RKHS norm is bounded by 1.

Assumption 2. *Assume that the transition probability distribution satisfies,*

$$\forall s' \in \mathcal{S}: \quad \|P(s' \mid \cdot,\cdot)\|_{\mathcal{H}_K} \le 1\,.$$

This assumption is very flexible given the generality of the RKHSs. This is a standard assumption which is also used in Yang et al. (2020b). We do not make any explicit assumptions on the Q-function related to the policy. Recall the definition of PV from Section 2.1. In Lemma 3, we prove that for any $V : \mathcal{S} \to [0, \frac{1}{1-\gamma}]$, $\|PV\|_{\mathcal{H}_K} \le \frac{c}{1-\gamma}$, as a consequence of Assumption 2, that is essential for our analysis.

[3] A random variable X is said to be sub-Gaussian with parameter R if its moment generating function is bounded by that of a zero mean Gaussian with variance R^2.

Some generic notation: For any real number x, and real numbers a, b, the notation $\Pi_{[a,b]}[x]$ is used to denote the projection of x onto $[a, b]$. For any integer J, I_J denotes the $J \times J$ identity matrix, and $\mathbf{0}_J$ denotes the $J \times 1$ zero vector.

3 KERNEL BASED Q-LEARNING

In this section, we present a novel kernel based Q-learning algorithm, referred to as KQLearn. Recall $Q(s, a) = r(s, a) + \gamma[PV](s, a)$. The transition probability distribution P and the value function V are both unknown to the algorithm. The algorithm, thus, recursively approximates PV, in rounds, using kernel ridge regression of PV from the observations in the previous round. I.e., the algorithm performs updates based on an approximate Bellman operator using predictions for PV provided by the kernel model. The samples are collected based on uncertainties for PV in the kernel model. For this purpose, the algorithm first creates a maximum uncertainty set which is used to collect the samples.

Maximum Uncertainty Set ($\mathcal{U}_J$): The algorithm starts with creating a maximum uncertainty set with size J referred to as $\mathcal{U}_J \subset \mathcal{Z}$. This set is created based on the uncertainties provided by the kernel model. In particular, each state-action is added to this set based on the following rule: choose the state-action with the highest uncertainty in the kernel model.

$$(s_j, a_j) = \underset{(s,a)\in\mathcal{Z}}{\arg\max}\, \Sigma^2_{\mathcal{U}_{j-1}}(s, a). \tag{6}$$

Then, recursively, $\mathcal{U}_j = \mathcal{U}_{j-1} \cup \{(s_j, a_j)\}$, starting from $\mathcal{U}_0 = \varnothing$. The set $\mathcal{U}_J$ is then used to collect samples from the generative model.

The algorithm proceeds in rounds indexed by $\ell = 1, \dots, L$. Each round ℓ receives noisy observations $Y^{(\ell-1)}_{\mathcal{U}_J}$ of PV from the previous round, $\ell-1$. These observations are then used within kernel ridge regression to form a regressor of PV over entire $\mathcal{Z}$, and obtain new observation $Y^{(\ell)}$. The observation vector is initialized to a zero vector $Y^{(0)} = \mathbf{0}_J$.

During each round ℓ, for each state-action pair $(s_j, a_j) \in \mathcal{U}_J$, a transition state $s'_j \sim P(\cdot|s_j, a_j)$ is acquired from the generative model. The observation $Y^{(\ell)}(s_j, a_j)$ is then given as follows:

$$\begin{aligned} Y^{(\ell)}(s_j, a_j) = \Pi_{[0, \frac{1}{1-\gamma}]} \max_{a\in\mathcal{A}} \Big\{ & r(s'_j, a) \\ & + \gamma k_{\mathcal{U}_J}(s'_j, a)^\top (K_{\mathcal{U}_J} + \lambda^2 I_J)^{-1} Y^{(\ell-1)}_{\mathcal{U}_J} \Big\}. \end{aligned} \tag{7}$$

The second term on the right hand side is the regressor in kernel ridge regression on PV, using $Y^{(\ell-1)}_{\mathcal{U}_J}$ as a vector of observations. In the analysis, we show a high probability bound on the error of this regression. The vector $Y^{(\ell)}_{\mathcal{U}_J} = [Y^{(\ell)}(s_1, a_1), \dots, Y^{(\ell)}(s_J, a_J)]^\top$ can be understood as updated noisy observations of PV which is passed to the next round, $\ell + 1$. By definition of the value function and the assumption of bounded rewards, it can be easily checked that $0 \le V^*(s) \le \frac{1}{1-\gamma}$, for all $s \in \mathcal{S}$. We thus project the value of PV on $[0, \frac{1}{1-\gamma}]$ interval.

KQLearn collects $N = JL$ samples in total.[4] A pseudocode is provided in Algorithm 1.

After collecting all samples, the KQLearn algorithm returns an RL policy π which selects the actions based on the following proxy Q-function:

$$\widehat{Q}^{(L)}(s, a) = r(s, a) + \gamma k^\top_{\mathcal{U}_J}(s, a)(K_{\mathcal{U}_J} + \lambda^2 I_J)^{-1} Y^{(L)}_{\mathcal{U}_J}. \tag{8}$$

Specifically, when state s is observed, the policy π selects the action $\pi(s) = \arg\max_{a\in\mathcal{A}} \widehat{Q}^{(L)}(s, a)$. The second term on the right hand side is the kernel ridge regression of PV using the observation in round L of the KQlearn algorithm.

Algorithm 1 Kernel-based Q-learning (KQLearn)

Input Discounted MDP with a generative model, kernel K, regularization parameter $\lambda > 0$, and $N > 0$
Output $\widehat{Q}^{(L)} : \mathcal{Z} \to \mathbb{R}$

```
Initialize L, J ∈ ℕ, N = LJ.
Initialize Y^(0) = 0_J and the set U_0 = ∅.
for all j = 1, ..., J do
    Update the function Σ_{U_{j-1}}(·) using Equation 2.
    Pick (s_j, a_j) ← argmax_{(s,a)∈Z} Σ²_{U_{j-1}}(s, a).
    U_j ← U_{j-1} ∪ {(s_j, a_j)}.
end for

for all ℓ = 1, ..., L do                                   ▷ round
    for all j = 1, ..., J do
        Obtain a sample transition state s' ~ P(·|s_j, a_j)
        Update the Y^(ℓ) as follows.
            Y^(ℓ)(s_j, a_j) ← Π_[0, 1/(1-γ)] max_{a∈A} {r(s', a)
                + γ k^T_{U_J}(s', a) (K_{U_J} + λ I_J)^{-1} Y^(ℓ-1)_{U_J}}
    end for
end for
Q̂^(L)(·) = r(·) + γ k^T_{U_J}(·)(K_{U_J} + λ² I_J)^{-1} Y^(L)_{U_J}.
```

[4] For the simplicity of presentation, we assume $N = JL$. When J does not divide N, we can ignore the samples in the last round.

4 SAMPLE COMPLEXITY OF KQLEARN

In this section, we present our theoretical results. The following theorem establishes a bound on the error in the value function for the policy obtained in the KQLearn algorithm.

Theorem 1. *Consider the discounted MDP described in Section 2.1. Consider the KQLearn algorithm described in Section 3. Under Assumptions 1 and 2, with probability at least* $1-\delta$,

$$\|V^\pi - V^*\|_\infty \leq \quad 2\beta(\delta)\left(\frac{\gamma}{1-\gamma}\right)^2\sqrt{\frac{2\Gamma_{K,\lambda}(J)}{J}} + 2\gamma^{L-1}\left(\frac{1}{1-\gamma}\right)^2,$$

where $\beta(\delta) = \mathcal{O}\left(\frac{c}{1-\gamma} + \frac{1}{2\lambda(1-\gamma)}\sqrt{d\log(\frac{J}{(1-\gamma)\delta})}\right)$ *and* c *is a constant given in Lemma 3.*

When J and L are sufficiently large, both terms in the upper bound become arbitrarily small.

4.1 Sample Complexity

We can obtain explicit sample complexities for the KQLearn algorithm, using kernel specific bounds on $\Gamma_{K,\lambda}$, which depend on the decay rate of the Mercer eigenvalues of K. In particular, we define the following characteristic eigendecay profiles (which are similar to those outlined in Chatterji et al. (2019); Vakili et al. (2021c); Yang et al. (2020b)).

Definition 2. *[Polynomial and Exponential Eigendecay] Consider the Mercer eigenvalues* $\{\sigma_m\}_{m=1}^\infty$ *of* K *as given in Equation 1 in a decreasing order.*

(i) For some $C_p > 0$, $\beta_p > 1$, K *is said to have a* (C_p, β_p) *polynomial eigendecay, if for all* $m \in \mathbb{N}$, *we have* $\sigma_m \leq C_p m^{-\beta_p}$.

(ii) For some $C_{e,1}, C_{e,2}, \beta_e > 0$, K *is said to have a* $(C_{e,1}, C_{e,2}, \beta_e)$ *exponential eigendecay, if for all* $m \in \mathbb{N}$, *we have* $\sigma_m \leq C_{e,1}\exp(-C_{e,2}m^{\beta_e})$.

We are now ready to present explicit bounds on the sample complexity for the very general classes of kernels with polynomial and exponential decay of Mercer eigenvalues.

Theorem 2. *Consider the discounted MDP described in Section 2.1. Consider the KQLearn algorithm described in Section 3, with* $L = \Theta\left(\frac{\log\left(\epsilon(1-\gamma)^2\right)}{1-\gamma}\right)$ *and* $J = \frac{N}{L}$. *Under Assumptions 1 and 2, KQLearn obtains an* ϵ*-optimal policy with probability at least* $1-\delta$, *with a sample complexity at most*

- *In the case of a kernel with* (C_p, β_p) *polynomial eigendecay,*

$$N = \tilde{\mathcal{O}}\left(\frac{\left(\log(\frac{1}{\delta})\right)^{\frac{\beta_p}{\beta_p-1}}}{\epsilon^{\frac{2\beta_p}{\beta_p-1}}(1-\gamma)^{\frac{7\beta_p-1}{\beta_p-1}}}\right). \tag{9}$$

- *In the case of a kernel with* $(C_{e,1}, C_{e,2}, \beta_e)$ *exponential eigendecay,*

$$N = \tilde{\mathcal{O}}\left(\frac{\log(\frac{1}{\delta})}{\epsilon^2(1-\gamma)^7}\right). \tag{10}$$

The detailed expressions for J and L including the implied constants and logarithmic factors in the $\tilde{\mathcal{O}}$ notation are provided in Appendix B, Equations 29 and 32.

Specific Kernels: Our bounds on the sample complexity can be specialized for various kernels where the eigendecay or bounds on $\Gamma_{K,\lambda}$ is known (such as the ones in Srinivas et al., 2010; Vakili et al., 2021c,a). Specifically, for the Matérn and SE kernels, we have, respectively,

$$N = \tilde{\mathcal{O}}\left(\frac{\left(\log(\frac{1}{\delta})\right)^{1+\frac{d}{2\nu}}}{\epsilon^{2+\frac{d}{\nu}}(1-\gamma)^{7+\frac{3d}{\nu}}}\right), \text{ and } N = \tilde{\mathcal{O}}\left(\frac{\log(\frac{1}{\delta})}{\epsilon^2(1-\gamma)^7}\right).$$

4.2 Optimality of the Sample Complexities

The sample complexities given above are order optimal with respect to ϵ. We compare them to the lower bounds on the sample complexity for kernel bandits (that is a special case of our setting when $|\mathcal{S}| = 1$). In particular, Scarlett et al. (2017) proved $\Omega\left((\frac{1}{\epsilon})^{2+\frac{d}{\nu}}\right)$ and $\Omega\left(\frac{1}{\epsilon^2}\right)$ sample complexities for the Matérn and SE kernels, respectively. Our results are the first finite sample complexities for the RL problem under a very general case which includes all kernels with polynomially decaying eigenvalues.

In terms of the discount factor, our sample complexities scale with $\frac{1}{(1-\gamma)^7}$ in the case of smooth kernels (similar to the PPQ-Learning algorithm, in the linear setting, Yang and Wang, 2019). In the tabular and linear settings, however, this has been improved to $\mathcal{O}(\frac{1}{(1-\gamma)^3})$. It remains an interesting problem for future investigation that whether the dependency of the sample complexity on the discount factor can be improved to $\mathcal{O}(\frac{1}{(1-\gamma)^3})$, also in the kernel setting. As discussed in the introduction, in the kernel setting, neither observing all state-actions nor a parametric update of the model through value iteration is feasible. Thus, a different approach to algorithm design and analysis is required that is increasingly more challenging among these settings: *tabular* $\rightarrow$ *linear* $\rightarrow$ *kernel-based.*

5 ANALYSIS

Theorem 2 is a consequence of Theorem 1, and using the kernel specific bounds on $\Gamma_{K,\lambda}$. The proof of Theorem 1 builds on several components including tracking the approximation error in kernel ridge regression and convergence error of an approximate Bellman operator. In this section, we overview the main steps in the proof of Theorem 1, while deferring the details to the appendix.

Approximate Bellman operator: Recall the Bellman operator defined in Section 2.1. The transition probability distribution and the value function are complex non-linear functions on continuous domains, unknown to the algorithm. We thus define an approximate Bellman operator $\widehat{\mathcal{T}}$, which uses noisy observations of PV on a fixed set $\mathcal{U}_J$, and takes advantage of kernel ridge regression, to perform an approximate Bellman operator update. In particular, for all $V : \mathcal{S} \to [0, \frac{1}{1-\gamma}]$, and a fixed set $\mathcal{U}_J \subset \mathcal{Z}$, let us define

$$[\widehat{\mathcal{T}}V](s) = \max_{a\in\mathcal{A}} \Big\{ r(s,a) + \gamma k^\top_{\mathcal{U}_J}(s,a)(K_{\mathcal{U}_J} + \lambda^2 I_J)^{-1}[\widehat{P}V]_{\mathcal{U}_J} \Big\}, \quad (11)$$

where $[\widehat{P}V](s,a) = V(s')$ is a random variable, $s' \sim P(\cdot|s,a)$ is a random transition state, and

$$[\widehat{P}V]_{\mathcal{U}_J} = \Big[[\widehat{P}V](s_1,a_1), \ldots, [\widehat{P}V](s_J,a_J) \Big]^\top .$$

In the KQLearn algorithm, define

$$\widehat{V}^{(\ell)}(s) = \max_{a\in\mathcal{A}} \Big\{ r(s,a) + \gamma k^\top_{\mathcal{U}_J}(s,a) \left(K_{\mathcal{U}_J} + \lambda^2 I_J\right)^{-1} Y^{(\ell)}_{\mathcal{U}_J} \Big\}, \quad (12)$$

which we refer to as proxy value function (similar to the proxy Q-function given in (8)). We then have the following recursive relation over ℓ.

$$\widehat{V}^{(\ell)} = \widehat{\mathcal{T}}\Pi_{[0,\frac{1}{1-\gamma}]}[\widehat{V}^{(\ell-1)}]. \quad (13)$$

Error in proxy value function: In order to bound the error in the value function of the policy π obtained by KQLearn, $\|V^\pi - V^*\|$, we need to bound the error in the proxy Q-function given in (8), which is used to obtain π. The error in proxy Q-function can be bounded based on the error in the proxy value function at round $L-1$. In particular, we have

$$\|\widehat{Q}^{(L)} - Q^*\|_\infty \le \|\widehat{V}^{(L-1)} - V^*\|_\infty. \quad (14)$$

Therefore, we next bound the error in the proxy value function. We can write the error in proxy value function as the sum of two terms: the error in approximate Bellman operator and the error in the value function using true Bellman operator. Specifically,

$$\begin{aligned}\left\|\widehat{V}^{(L-1)} - V^*\right\|_\infty &= \left\|\widehat{\mathcal{T}}\Pi_{[0,\frac{1}{1-\gamma}]}\widehat{V}^{(L-2)} - V^*\right\|_\infty \\ &\le \left\|\widehat{\mathcal{T}}\Pi_{[0,\frac{1}{1-\gamma}]}\widehat{V}^{(L-2)} - \mathcal{T}\Pi_{[0,\frac{1}{1-\gamma}]}\widehat{V}^{(L-2)}\right\|_\infty \\ &+ \left\|\mathcal{T}\Pi_{[0,\frac{1}{1-\gamma}]}\widehat{V}^{(L-2)} - V^*\right\|_\infty . \end{aligned} \quad (15)$$

The second term can be recursively bounded which leads to the second term $\frac{2\gamma^{L-1}}{(1-\gamma)^2}$ in the error bound in Theorem 1. The first term leads to an important step in the analysis which is based on the error in kernel ridge regression. Specifically, let $V : \mathcal{S} \to [0, \frac{1}{1-\gamma}]$ be any value function. We have, for all $s \in \mathcal{S}$,

$$\begin{aligned}&[\widehat{\mathcal{T}}V](s) - [\mathcal{T}V](s) \\ &= \max_{a\in\mathcal{A}} \Big\{ r(s,a) + \gamma k^\top_{\mathcal{U}_J}(s,a)(K_{\mathcal{U}_J} + \lambda^2 I_J)^{-1}[\widehat{P}V]_{\mathcal{U}_J} \Big\} \\ &\quad - \max_{a\in\mathcal{A}} \{ r(s,a) + \gamma[PV](s,a)\} \\ &\le \gamma \max_{a\in\mathcal{A}} \Big\{ k^\top_{\mathcal{U}_J}(s,a)(K_{\mathcal{U}_J} + \lambda^2 I_J)^{-1}[\widehat{P}V]_{\mathcal{U}_J} \\ &\quad - [PV](s,a) \Big\}. \end{aligned} \quad (16)$$

Error in kernel ridge regression: The term inside max in Equation 16 is the error in kernel ridge regression, where PV is the target function, $[\widehat{P}V]_{\mathcal{U}_J}$ is a set of J noisy observations, and $k^\top_{\mathcal{U}_J}(s,a)(K_{\mathcal{U}_J} + \lambda^2 I_J)^{-1}[\widehat{P}V]_{\mathcal{U}_J}$ is the regressor. In order to apply Lemma 1, we need an upper bound on the RKHS norm of PV, as well as an upper bound on the sub-Gaussianity parameter of the observation noise in $\widehat{P}V$. These are established in the following lemmas.

Lemma 3. *Consider an integrable value function $V : \mathcal{S} \to [0, \frac{1}{1-\gamma}]$. Under Assumption 2, we have*

$$\|PV\|_{\mathcal{H}_K} \le \frac{c}{1-\gamma}, \quad (17)$$

where $c = \int_{\mathcal{S}} ds$ is the volume of $\mathcal{S}$.

Lemma 4. *Consider a transition probability distribution P, and an integrable value function $V : \mathcal{S} \to [0, \frac{1}{1-\gamma}]$. We have, for all (s,a), $\mathbb{E}\Big[[\widehat{P}V](s,a)\Big] = PV(s,a)$. In addition, $[\widehat{P}V](s,a)$ is a sub-Gaussian random variable with parameter $\frac{1}{2(1-\gamma)}$.*

Lemma 4 follows from the definition of $\widehat{P}V$, as well as Hoeffding lemma for bounded random variables. A proof of Lemma 3 is provided in Appendix C.

Applying Lemma 1, we obtain, with probability $1-\delta$, for all $(s,a)\in\mathcal{Z}$,

$$\begin{aligned}\left|k_{\mathcal{U}_J}^\top(s,a)(K_{\mathcal{U}_J}+\lambda^2 I_J)^{-1}[\widehat{P}V]_{\mathcal{U}_J}-[PV](s,a)\right|\\ \leq \beta(\delta)\Sigma_{\mathcal{U}_J}(s,a),\end{aligned} \tag{18}$$

where $\beta(\delta)=\mathcal{O}\left(\frac{c}{1-\gamma}+\frac{1}{2(1-\gamma)\lambda}\sqrt{d\log(\frac{Jc}{(1-\gamma)\delta})}\right)$.

Eventually, using Lemma 2 on the total uncertainty, and by the design of $\mathcal{U}_J$, we bound $\Sigma_{\mathcal{U}_J}(s,a)$ on the right hand side.

We thus bounded the two terms in (21), that bounds the error in proxy value function. More details on the proof of theorems and the proof of lemmas are provided in Appendix B and Appendix C, respectively.

6 CONCLUSION

Modern RL often faces an enormous state-action space and complex models. We considered the question of sample complexity in a discounted MDP with a generative model under the kernel setting, furthering a line of research in the literature (e.g., see Kearns and Singh, 1998; Azar et al., 2017; Sidford et al., 2018a,b; Yang and Wang, 2019). We introduced a novel kernel-based Q learning algorithm referred to as KQLearn and proved a finite bound on its sample complexity for very general classes of kernels. That is to the best of our knowledge the first finite sample complexity result under the general kernel setting (including all kernels with polynomially decaying eigenvalues). In addition, compared to the lower bounds on the special case of the kernel bandit problem, our sample complexities are tight with respect to ϵ in finding an ϵ-optimal policy. Our sample complexities, however, scale possibly suboptimally with respect to the discount factor, which remains an interesting open problem for future investigation.

References

Abbasi-Yadkori, Y. (2013). Online learning for linearly parametrized control problems.

Agarwal, A., Jiang, N., Kakade, S. M., and Sun, W. (2019). Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, pages 10–4.

Agarwal, A., Kakade, S., and Yang, L. F. (2020). Model-based reinforcement learning with a generative model is minimax optimal. In *Conference on Learning Theory*, pages 67–83. PMLR.

Ayoub, A., Jia, Z., Szepesvari, C., Wang, M., and Yang, L. (2020). Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, pages 463–474. PMLR.

Azar, M. G., Munos, R., and Kappen, H. J. (2013). Minimax pac bounds on the sample complexity of reinforcement learning with a generative model. *Machine learning*, 91(3):325–349.

Azar, M. G., Osband, I., and Munos, R. (2017). Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pages 263–272. PMLR.

Carpentier, A., Vernade, C., and Abbasi-Yadkori, Y. (2020). The elliptical potential lemma revisited. *arXiv preprint arXiv:2010.10182*.

Chatterji, N., Pacchiano, A., and Bartlett, P. (2019). Online learning with kernel losses. In *Proceedings of Machine Learning Research*, volume 97, pages 971–980, Long Beach, California, USA. PMLR.

Chowdhury, S. R. and Gopalan, A. (2017). On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pages 844–853. PMLR.

Christmann, A. and Steinwart, I. (2008). *Support Vector Machines*. Springer New York, NY.

Domingues, O. D., Ménard, P., Pirotta, M., Kaufmann, E., and Valko, M. (2021). Kernel-based reinforcement learning: A finite-time analysis. In *International Conference on Machine Learning*, pages 2783–2792. PMLR.

Fawzi, A., Balog, M., Huang, A., Hubert, T., Romera-Paredes, B., Barekatain, M., Novikov, A., R Ruiz, F. J., Schrittwieser, J., Swirszcz, G., et al. (2022). Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. (2018). Is q-learning provably efficient? *Advances in Neural Information Processing Systems*, 31.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. (2020). Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR.

Kahn, G., Villaflor, A., Pong, V., Abbeel, P., and Levine, S. (2017). Uncertainty-aware reinforcement learning for collision avoidance. *arXiv preprint arXiv:1702.01182*.

Kakade, S., Krishnamurthy, A., Lowrey, K., Ohnishi, M., and Sun, W. (2020). Information theoretic regret bounds for online nonlinear control. *Advances in Neural Information Processing Systems*, 33:15312–15325.

Kakade, S. M. (2003). *On the sample complexity of reinforcement learning*. University of London, University College London (United Kingdom).

Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., Quillen, D., Holly, E., Kalakrishnan, M., Vanhoucke, V., et al. (2018). Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, pages 651–673. PMLR.

Kearns, M. and Singh, S. (1998). Finite-sample convergence rates for q-learning and indirect algorithms. In *Advances in Neural Information Processing Systems*, volume 11. MIT Press.

Lee, K., Kim, S.-A., Choi, J., and Lee, S.-W. (2018). Deep reinforcement learning in continuous action spaces: a case study in the game of simulated curling. In *International Conference on Machine Learning,*, pages 2937–2946. PMLR.

Li, Z. and Scarlett, J. (2022). Gaussian process bandit optimization with few batches. In *International Conference on Artificial Intelligence and Statistics*, pages 92–107. PMLR.

Mercer, J. (1909). Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 209:415–446.

Mirhoseini, A., Goldie, A., Yazgan, M., Jiang, J. W., Songhori, E., Wang, S., Lee, Y.-J., Johnson, E., Pathak, O., Nazi, A., et al. (2021). A graph placement methodology for fast chip design. *Nature*, 594(7862):207–212.

Ormoneit, D. and Sen, A. (2002). Kernel-based reinforcement learning. *Machine learning*, 49(2-3):161.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Russo, D. (2019). Worst-case regret bounds for exploration via randomized value functions. *Advances in Neural Information Processing Systems*, 32.

Salgia, S., Vakili, S., and Zhao, Q. (2021). A domain-shrinking based bayesian optimization algorithm with order-optimal regret performance. *Advances in Neural Information Processing Systems*, 34:28836–28847.

Scarlett, J., Bogunovic, I., and Cevher, V. (2017). Lower bounds on regret for noisy gaussian process bandit optimization. In *Conference on Learning Theory*, pages 1723–1742. PMLR.

Schölkopf, B., Smola, A. J., Bach, F., et al. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. (2015). Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.

Sidford, A., Wang, M., Wu, X., Yang, L., and Ye, Y. (2018a). Near-optimal time and sample complexities for solving markov decision processes with a generative model. *Advances in Neural Information Processing Systems*, 31.

Sidford, A., Wang, M., Wu, X., and Ye, Y. (2018b). Variance reduced value iteration and faster algorithms for solving markov decision processes. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 770–787. SIAM.

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

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. In *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*, pages 1015–1022.

Vakili, S., Bouziani, N., Jalali, S., Bernacchia, A., and Shiu, D.-S. (2021a). Optimal order simple regret for gaussian process bandits. *Advances in Neural Information Processing Systems*, 34:21202–21215.

Vakili, S., Bromberg, M., Garcia, J., Shiu, D.-s., and Bernacchia, A. (2021b). Uniform generalization bounds for overparameterized neural networks. *arXiv preprint arXiv:2109.06099*.

Vakili, S., Khezeli, K., and Picheny, V. (2021c). On information gain and regret bounds in gaussian process bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 82–90. PMLR.

Vakili, S., Scarlett, J., and Javidi, T. (2021d). Open problem: Tight online confidence intervals for rkhs elements. In *Conference on Learning Theory*, pages 4647–4652. PMLR.

Vakili, S., Scarlett, J., Shiu, D.-S., and Bernacchia, A. (2022). Improved convergence rates for sparse approximation methods in kernel-based learning. *arXiv preprint arXiv:2202.04005*.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354.

Wang, R., Salakhutdinov, R., and Yang, L. F. (2020). Provably efficient reinforcement learning with general value function approximation. *arXiv preprint arXiv:2005.10804*.

Yang, L. and Wang, M. (2019). Sample-optimal parametric q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004. PMLR.

Yang, L. and Wang, M. (2020). Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pages 10746–10756. PMLR.

Yang, Z., Jin, C., Wang, Z., Wang, M., and Jordan, M. (2020a). Provably efficient reinforcement learning with kernel and neural function approximations. *Advances in Neural Information Processing Systems*, 33:13903–13916.

Yang, Z., Jin, C., Wang, Z., Wang, M., and Jordan, M. I. (2020b). On function approximation in reinforcement learning: Optimism in the face of large state spaces. *arXiv preprint arXiv:2011.04622*.

Zhou, D., He, J., and Gu, Q. (2021). Provably efficient reinforcement learning for discounted mdps with feature mapping. In *International Conference on Machine Learning*, pages 12793–12802. PMLR.

In the appendix, we provide some details and proofs omitted from the main paper due to space limit. In Appendix A, we provide a formal statement of Mercer theorem and a constructive definition of the RKHS. The proof of Theorems 1 and 2, and auxiliary lemmas are provided in Appendix B and Appendix C, respectively.

A MERCER THEOREM

Mercer theorem (Mercer, 1909) provides a representation of the kernel in terms of an infinite dimensional feature map (see, e.g. Christmann and Steinwart (2008), Theorem 4.49). Let $\mathcal{Z}$ be a compact metric space and μ be a finite Borel measure on $\mathcal{Z}$ (we consider Lebesgue measure in a Euclidean space). Let $L^2_\mu(\mathcal{Z})$ be the set of square-integrable functions on $\mathcal{Z}$ with respect to μ. We further say a kernel is square-integrable if

$$\int_{\mathcal{Z}}\int_{\mathcal{Z}} K(z,z')^2\, d\mu(z)d\mu(z') < \infty.$$

Theorem 3. *(Mercer Theorem) Let $\mathcal{Z}$ be a compact metric space and μ be a finite Borel measure on $\mathcal{Z}$. Let K be a continuous and square-integrable kernel, inducing an integral operator $T_K : L^2_\mu(\mathcal{Z}) \to L^2_\mu(\mathcal{Z})$ defined by*

$$\left(T_K f\right)(\cdot) = \int_{\mathcal{Z}} K(\cdot, z') f(z')\, d\mu(z')\,,$$

where $f \in L^2_\mu(\mathcal{Z})$. Then, there exists a sequence of eigenvalue-eigenfunction pairs $\{(\sigma_m, \psi_m)\}_{m=1}^\infty$ such that $\sigma_m > 0$, and $T_K\psi_m = \sigma_m\psi_m$, for $m \geq 1$. Moreover, the kernel function can be represented as

$$K\left(z, z'\right) = \sum_{m=1}^{\infty} \sigma_m \psi_m(z) \psi_m\left(z'\right),$$

where the convergence of the series holds uniformly on $\mathcal{Z} \times \mathcal{Z}$.

According to Mercer representation theorem (see, e.g., Christmann and Steinwart (2008), Theorem 4.51), the RKHS induced by K can consequently be represented in terms of $\{(\sigma_m, \psi_m)\}_{m=1}^\infty$.

Theorem 4. *(Mercer Representation Theorem) Let $\{(\sigma_m, \psi_m)\}_{i=1}^\infty$ be the Mercer eigenvalue eigenfunction pairs. Then, the RKHS of K is given by*

$$\mathcal{H}_K = \left\{ f(\cdot) = \sum_{i=1}^{\infty} w_i \sigma_i^{\frac{1}{2}} \psi_i(\cdot) : w_i \in \mathbb{R}, \|f\|^2_{\mathcal{H}_K} := \sum_{i=1}^{\infty} w_i^2 < \infty \right\}.$$

Mercer representation theorem indicates that the scaled eigenfunctions $\{\sqrt{\sigma_i}\psi_i\}_{i=1}^\infty$ form an orthonormal basis for $\mathcal{H}_K$.

B PROOF OF THEOREMS

In this section, we provide the proof of main theorems.

B.1 Proof of Theorem 1.

The proof of Theorem 1 builds on an approximate Bellman operator, that uses noisy observations of PV within the rounds of KQLearn, and kernel ridge regression. We prove bounds on the error of this approximate Bellman operator. That is then used to bound the error in the value function of the policy obtained by KQLearn.

Approximate Bellman operator: Recall the Bellman operator defined in Section 2.1. The transition probability distribution and the value function are complex non-linear functions on continuous domains, unknown to the algorithm. We thus define an approximate Bellman operator $\widehat{\mathcal{T}}$, which uses noisy observations of PV on a fixed set $\mathcal{U}_J$, and takes advantage of kernel ridge regression, to perform an approximate Bellman operator update. In particular, for all $V : \mathcal{S} \to [0, \frac{1}{1-\gamma}]$, and a fixed set $\mathcal{U}_J \subset \mathcal{Z}$, let us define

$$[\widehat{\mathcal{T}}V](s) = \max_{a\in\mathcal{A}} \left\{ r(s,a) + \gamma k^\top_{\mathcal{U}_J}(s,a)(K_{\mathcal{U}_J} + \lambda^2 I_J)^{-1}[\widehat{P}V]_{\mathcal{U}_J} \right\},$$

where $[\widehat{P}V](s,a) = V(s')$ is a random variable, $s' \sim P(\cdot|s,a)$ is a random transition state, and

$$[\widehat{P}V]_{\mathcal{U}_J} = \left[[\widehat{P}V](s_1,a_1),\dots,[\widehat{P}V](s_J,a_J)\right]^\top.$$

In the KQLearn algorithm, define

$$\widehat{V}^{(\ell)}(s) = \max_{a\in\mathcal{A}}\left\{r(s,a) + \gamma k_{\mathcal{U}_J}^\top(s,a)\left(K_{\mathcal{U}_J}+\lambda^2 I_J\right)^{-1} Y_{\mathcal{U}_J}^{(\ell)}]\right\},$$

which we refer to as proxy value function (similar to the proxy Q-function given in (8)). We then have the following recursive relation over ℓ.

$$\widehat{V}^{(\ell)} = \widehat{\mathcal{T}}\Pi_{[0,\frac{1}{1-\gamma}]}[\widehat{V}^{(\ell-1)}]. \tag{19}$$

Error in proxy value function: We next bound the error in the proxy value function. We can write the error in proxy value function as the sum of two terms: the error in approximate Bellman operator and the error in the value function using true Bellman operator. Specifically, for $l > 1$,

$$\begin{aligned}\left\|\widehat{V}^{(l-1)} - V^*\right\|_\infty &= \left\|\widehat{\mathcal{T}}\Pi_{[0,\frac{1}{1-\gamma}]}\widehat{V}^{(l-2)} - V^*\right\|_\infty \\ &\le \underbrace{\left\|\widehat{\mathcal{T}}\Pi_{[0,\frac{1}{1-\gamma}]}\widehat{V}^{(l-2)} - \mathcal{T}\Pi_{[0,\frac{1}{1-\gamma}]}\widehat{V}^{(l-2)}\right\|_\infty}_{\text{Term I}} \\ &\quad + \underbrace{\left\|\mathcal{T}\Pi_{[0,\frac{1}{1-\gamma}]}\widehat{V}^{(l-2)} - V^*\right\|_\infty}_{\text{Term II}}.\end{aligned} \tag{20}$$

We now bound the two terms on the right hand side of (20).

Term I: The first term leads us to an important step in the analysis which is based on the error in kernel ridge regression. Specifically, let $V : \mathcal{S} \to [0, \frac{1}{1-\gamma}]$ be any value function. We have, for all $s \in \mathcal{S}$,

$$\begin{aligned}[\widehat{\mathcal{T}}V](s) - [\mathcal{T}V](s) &= \max_{a\in\mathcal{A}}\left\{r(s,a) + \gamma k_{\mathcal{U}_J}^\top(s,a)(K_{\mathcal{U}_J}+\lambda^2 I_J)^{-1}[\widehat{P}V]_{\mathcal{U}_J}\right\} - \max_{a\in\mathcal{A}}\{r(s,a)+\gamma[PV](s,a)\} \\ &\le \gamma\max_{a\in\mathcal{A}}\left\{k_{\mathcal{U}_J}^\top(s,a)(K_{\mathcal{U}_J}+\lambda^2 I_J)^{-1}[\widehat{P}V]_{\mathcal{U}_J} - [PV](s,a)\right\}.\end{aligned} \tag{21}$$

Error in kernel ridge regression: The term inside $\max$ in Equation 21 is the error in kernel ridge regression, where PV is the target function, $[\widehat{P}V]_{\mathcal{U}_J}$ is a set of J noisy observations, and $k_{\mathcal{U}_J}^\top(s,a)(K_{\mathcal{U}_J}+\lambda^2 I_J)^{-1}[\widehat{P}V]_{\mathcal{U}_J}$ is the regressor. In order to apply Lemma 1, we need an upper bound on the RKHS norm of PV, as well as an upper bound on the sub-Gaussianity parameter of the observation noise in $\widehat{P}V$. These are established in Lemmas 3 and 4, respectively. Specifically, we have

$$\|PV\|_{\mathcal{H}_K} \le \frac{c}{1-\gamma}. \tag{22}$$

And, $[\widehat{P}V](s,a)$ is a sub-Gaussian random variable with parameter $\frac{1}{2(1-\gamma)}$.

Applying Lemma 1, we obtain, with probability $1-\delta$, for all $(s,a) \in \mathcal{Z}$,

$$\left|k_{\mathcal{U}_J}^\top(s,a)(K_{\mathcal{U}_J}+\lambda^2 I_J)^{-1}[\widehat{P}V]_{\mathcal{U}_J} - [PV](s,a)\right| \le \beta(\delta)\Sigma_{\mathcal{U}_J}(s,a), \tag{23}$$

where $\beta(\delta) = \mathcal{O}\left(\frac{c}{1-\gamma} + \frac{1}{2(1-\gamma)\lambda}\sqrt{d\log(\frac{Jc}{(1-\gamma)\delta})}\right)$.

Bounding $\Sigma_{\mathcal{U}_J}(s,a)$: Conditioning on a smaller subset of observation reduces the variance $\Sigma_{\mathcal{U}_j}(s,a) \geq \Sigma_{\mathcal{U}_J}(s,a)$, for all $j \leq J$ (due to positive definiteness of the kernel matrix). By the selection rule of the observation points:

$$(s_j, a_j) = \underset{(s,a)\in\mathcal{Z}}{\arg\max}\, \Sigma^2_{\mathcal{U}_{j-1}}(s,a), \tag{24}$$

we have $\Sigma_{\mathcal{U}_{j-1}}(s,a) \leq \Sigma_{\mathcal{U}_{j-1}}(s_j,a_j)$. Thus, for all $(s,a)\in\mathcal{Z}$,

$$\begin{aligned}
\Sigma^2_{\mathcal{U}_J}(s,a) &\leq \frac{1}{J}\sum_{j=1}^{J}\Sigma^2_{\mathcal{U}_{j-1}}(s,a)\\
&\leq \frac{1}{J}\sum_{j=1}^{J}\Sigma^2_{\mathcal{U}_{j-1}}(s_j,a_j)\\
&\leq \frac{2\Gamma_{K,\lambda}(J)}{\log(1+1/\lambda^2)J},
\end{aligned}$$

where the last line follows from Lemma 2.

Replacing the bound on $\Sigma_{\mathcal{U}_J}(s,a)$, we obtain, for all $V:\mathcal{S}\to[0,\frac{1}{1-\gamma}]$, $s\in\mathcal{S}$,

$$[\widehat{\mathcal{T}}V](s) - [\mathcal{T}V](s) \leq \gamma\beta(\delta)\sqrt{\frac{2\Gamma_{K,\lambda}(J)}{\log(1+1/\lambda^2)J}}. \tag{25}$$

Thus,

$$\left\|\widehat{\mathcal{T}}\Pi_{[0,\frac{1}{1-\gamma}]}\widehat{V}^{(l-2)} - \mathcal{T}\Pi_{[0,\frac{1}{1-\gamma}]}\widehat{V}^{(l-2)}\right\|_\infty \leq \gamma\beta(\delta)\sqrt{\frac{2\Gamma_{K,\lambda}(J)}{\log(1+1/\lambda^2)J}}. \tag{26}$$

Term II: We now bound the second term on the right hand side of (20), by the contraction of Bellman operator.

$$\begin{aligned}
\left\|\mathcal{T}\Pi_{[0,\frac{1}{1-\gamma}]}\widehat{V}^{(l-2)} - V^*\right\|_\infty &\leq \gamma\left\|\Pi_{[0,\frac{1}{1-\gamma}]}\widehat{V}^{(l-2)} - V^*\right\|_\infty\\
&\leq \gamma\left\|\widehat{V}^{(l-2)} - V^*\right\|_\infty.
\end{aligned} \tag{27}$$

The second inequality follows from the observation that $V^*(s)\in[0,\frac{1}{1-\gamma}]$ for all $s\in\mathcal{S}$.

Combing the bounds on Term I and Term II, we obtain

$$\left\|\widehat{V}^{(l-1)} - V^*\right\|_\infty \leq \gamma\beta(\delta)\sqrt{\frac{2\Gamma_{K,\lambda}(J)}{\log(1+1/\lambda^2)J}} + \gamma\left\|\widehat{V}^{(l-2)} - V^*\right\|_\infty.$$

Recursively bounding the error in proxy value function at round l using the error at round $l-1$ for $l=2,\dots,L-1$, we have,

$$\begin{aligned}
\left\|\widehat{V}^{(L-1)} - V^*\right\|_\infty &\leq \beta(\delta)\sqrt{\frac{2\Gamma_{K,\lambda}(J)}{\log(1+1/\lambda^2)J}}\left(\sum_{i=1}^{L-1}\gamma^i\right) + \gamma^{L-1}\left\|\widehat{V}^{(0)} - V^*\right\|_\infty\\
&\leq \beta(\delta)\sqrt{\frac{2\Gamma_{K,\lambda}(J)}{\log(1+1/\lambda^2)J}}\left(\sum_{i=1}^{L-1}\gamma^i\right) + \frac{\gamma^{L-1}}{1-\gamma},
\end{aligned}$$

where the second inequality comes from $\left\|\widehat{V}^{(0)} - V^*\right\|_\infty \leq \frac{1}{1-\gamma}$.

Recall the definition of the proxy Q-function $\widehat{Q}^{(L)}$ given in (8). We bound the error in $\widehat{Q}^{(L)}$ as follows. For all $(s,a) \in \mathcal{Z}$,

$$\begin{aligned}
\left\|\widehat{Q}^{(L)}(s,a) - Q^*(s,a)\right\|_\infty &= \left\|\left(r(s,a) + \gamma k_{\mathcal{U}_J}^\top(s,a)(K_{\mathcal{U}_J} + \lambda^2 I_J)^{-1} Y_{\mathcal{U}_J}^{(L)}\right) - \left(r(s,a) + \gamma[PV^*](s,a)\right)\right\|_\infty \\
&\leq \left\|\gamma\left(k_{\mathcal{U}_J}^\top(s,a)(K_{\mathcal{U}_J} + \lambda^2 I_J)^{-1}[\widehat{P}\Pi_{[0,\frac{1}{1-\gamma}]}\widehat{V}^{(L-1)}]_{\mathcal{U}_J} - [P\Pi_{[0,\frac{1}{1-\gamma}]}\widehat{V}^{(L-1)}](s,a)\right)\right\|_\infty \\
&\quad + \left\|\gamma\left([P\Pi_{[0,\frac{1}{1-\gamma}]}\widehat{V}^{(L-1)}](s,a) - [PV^*](s,a)\right)\right\|_\infty \\
&\leq \gamma\beta(\delta)\sqrt{\frac{2\Gamma_{K,\lambda}(J)}{\log(1+1/\lambda^2)J}} + \gamma\left\|\mathbb{E}_{s'\sim P(\cdot|s,a)}[\Pi_{[0,\frac{1}{1-\gamma}]}\widehat{V}^{(L-1)}] - \mathbb{E}_{s'\sim P(\cdot|s,a)}[V^*]\right\|_\infty \\
&\leq \gamma\beta(\delta)\sqrt{\frac{2\Gamma_{K,\lambda}(J)}{\log(1+1/\lambda^2)J}} + \gamma\left\|\Pi_{[0,\frac{1}{1-\gamma}]}\widehat{V}^{(L-1)} - V^*\right\|_\infty \\
&\leq \gamma\beta(\delta)\sqrt{\frac{2\Gamma_{K,\lambda}(J)}{\log(1+1/\lambda^2)J}} + \gamma\left\|\widehat{V}^{(L-1)} - V^*\right\|_\infty \\
&\leq \gamma\beta(\delta)\sqrt{\frac{2\Gamma_{K,\lambda}(J)}{\log(1+1/\lambda^2)J}} + \gamma\left(\beta(\delta)\sqrt{\frac{2\Gamma_{K,\lambda}(J)}{\log(1+1/\lambda^2)J}}\left(\sum_{i=1}^{L-1}\gamma^i\right) + \frac{\gamma^{L-1}}{1-\gamma}\right) \\
&= \beta(\delta)\sqrt{\frac{2\Gamma_{K,\lambda}(J)}{\log(1+1/\lambda^2)J}}\left(\sum_{i=1}^{L}\gamma^i\right) + \frac{\gamma^L}{1-\gamma} \\
&\leq \frac{\gamma\beta(\delta)}{1-\gamma}\sqrt{\frac{2\Gamma_{K,\lambda}(J)}{\log(1+1/\lambda^2)J}} + \frac{\gamma^L}{1-\gamma}.
\end{aligned}$$

The first inequality is a result of triangle inequality after adding and subtracting the term $\gamma[P\Pi_{[0,\frac{1}{1-\gamma}]}\widehat{V}^{(L-1)}](s,a)$. The second inequality is the error in kernel ridge regression bounded above in Term I. The third inequality bounds the difference in expectations with maximum difference. The fourth inequality is a consequence of the observation that $V^* \in [0, \frac{1}{1-\gamma}]$ interval. The fifth inequality is obtained using the bound on the error in the proxy value function given above.

The value function of π obtained from the proxy Q-function: The following lemma establishes that the error in the value function of π can be bounded using the error in the proxy Q-function.

Lemma 5. *Consider any Q-function $\tilde{Q}: \mathcal{S}\times\mathcal{A} \to [0, \frac{1}{1-\gamma}+1]$ satisfying $\|\tilde{Q} - Q^*\|_\infty \leq \epsilon$. Define policy $\pi_{\tilde{Q}}$ such that $\pi_{\tilde{Q}}(s) = \arg\max_{a\in\mathcal{A}} \tilde{Q}(s,a)$. Then, for all $s \in \mathcal{S}$,*

$$V^*(s) - V^{\pi_{\tilde{Q}}}(s) \leq \frac{2\epsilon}{1-\gamma}.$$

Applying Lemma 5 to $\widehat{Q}$ returned by KQLearn, it follows that with probability $1-\delta$,

$$\|V^\pi - V^*\|_\infty \leq \frac{2\gamma\beta(\delta)}{(1-\gamma)^2}\sqrt{\frac{2\Gamma_{K,\lambda}(J)}{\log(1+1/\lambda^2)J}} + \frac{2\gamma^L}{(1-\gamma)^2}.$$

That completes the proof.

B.2 Proof of Theorem 2

Theorem 2 is a consequence of Theorem 1 and the kernel specific bounds on $\Gamma_{K,\lambda}$. Several works have established bounds on $\Gamma_{K,\lambda}$ for various kernels (Srinivas et al., 2010; Vakili et al., 2021c,b). We use the reuslt in Vakili et al. (2021c) for kernels with polynomial and exponential eigendecay.

Polynomial Eigendecay: Consider a kernel with (C_p, β_p) polynomial eigendecay. We have the following bound on $\Gamma_{K,\lambda}$ (Vakili et al., 2021c). For all $J \in \mathbb{N}$,

$$\Gamma_{K,\lambda}(J) = \mathcal{O}(J^{\frac{1}{\beta_p}}\log^{1-\frac{1}{\beta_p}}(J)). \tag{28}$$

We replace this bound in the error in the value function of π obtained by KQLearn, in Theorem 1

$$\|V^\pi - V^*\|_\infty \leq \frac{2\gamma\beta(\delta)}{(1-\gamma)^2}\sqrt{\frac{2\Gamma_{K,\lambda}(J)}{\log(1+\frac{1}{\lambda^2})J}} + \frac{2\gamma^L}{(1-\gamma)^2}.$$

We then obtain

$$\|V^\pi - V^*\|_\infty = \mathcal{O}\left(\frac{\gamma}{(1-\gamma)^3}\left(c+\frac{1}{\lambda}\sqrt{d\log(\frac{Jc}{(1-\gamma)\delta})}\right)\sqrt{J^{\frac{1}{\beta_p}-1}\log^{1-\frac{1}{\beta_p}}(J)}\right) + \mathcal{O}\left(\frac{\gamma^L}{(1-\gamma)^2}\right).$$

We choose J and L large enough so that each term on the right hand side is bounded by $\epsilon/2$. The choices of

$$\begin{aligned} J &= \mathcal{O}\left(\left(\frac{1}{\epsilon}\right)^{\frac{2\beta_p}{\beta_p-1}}\frac{\gamma^{\frac{2\beta_p}{\beta_p-1}}}{(1-\gamma)^{\frac{6\beta_p}{\beta_p-1}}}\left(c+\frac{1}{\lambda}\sqrt{d\log(\frac{c}{(1-\gamma)\delta})}\right)^{\frac{2\beta_p}{\beta_p-1}}\log^{\frac{\beta_p}{\beta_p-1}}\left(\frac{1}{\epsilon(1-\gamma)}\right)\right), \\ L &= \mathcal{O}\left(\frac{\log(\frac{1}{\epsilon})+\log(\frac{1}{1-\gamma})}{(1-\gamma)}\right), \end{aligned} \tag{29}$$

with proper constants ensures $\|V^\pi - V^*\|_\infty \leq \epsilon$. The expression can be simplified as $J = \tilde{\mathcal{O}}\left(\frac{\left(\log(\frac{1}{\delta})\right)^{\frac{\beta_p}{\beta_p-1}}}{\epsilon^{\frac{2\beta_p}{\beta_p-1}}(1-\gamma)^{\frac{6\beta_p}{\beta_p-1}}}\right)$ and $L = \tilde{\mathcal{O}}\left(\frac{1}{1-\gamma}\right)$, omitting the logarithmic and constant terms. That leads to

$$N = \tilde{\mathcal{O}}\left(\frac{\left(\log(\frac{1}{\delta})\right)^{\frac{\beta_p}{\beta_p-1}}}{\epsilon^{\frac{2\beta_p}{\beta_p-1}}(1-\gamma)^{\frac{7\beta_p-1}{\beta_p-1}}}\right). \tag{30}$$

Exponential Eigendecay: Consider a kernel with $(C_{e_1}, C_{e_2}, \beta_e)$ polynomial eigendecay. We have the following bound on $\Gamma_{K,\lambda}$. For all $J \in \mathbb{N}$,

$$\Gamma_{K,\lambda}(J) = \mathcal{O}(\log^{1+\frac{1}{\beta_e}}(J)). \tag{31}$$

We replace this bound in the error in the value function of π obtained by KQLearn, in Theorem 1, and obtain

$$\|V^\pi - V^*\|_\infty = \mathcal{O}\left(\frac{\gamma}{(1-\gamma)^3}\left(c+\frac{1}{\lambda}\sqrt{d\log(\frac{Jc}{(1-\gamma)\delta})}\right)\sqrt{\frac{\log^{1+\frac{1}{\beta_e}}(J)}{J}}\right) + \mathcal{O}\left(\frac{\gamma^{L-1}}{(1-\gamma)^2}\right).$$

We choose J and L large enough so that each term on the right hand side is bounded by $\epsilon/2$. The choices of

$$\begin{aligned} J &= \mathcal{O}\left(\left(\frac{1}{\epsilon}\right)^2\frac{\gamma^2}{(1-\gamma)^6}\left(c+\frac{1}{\lambda}\sqrt{d\log(\frac{c}{(1-\gamma)\delta})}\right)^2\log^{2+\frac{1}{\beta_e}}\left(\frac{1}{\epsilon(1-\gamma)}\right)\right), \\ L &= \mathcal{O}\left(\frac{\log(\frac{1}{\epsilon})+\log(\frac{1}{1-\gamma})}{(1-\gamma)}\right), \end{aligned} \tag{32}$$

with proper constants ensures $\|V^\pi - V^*\|_\infty \leq \epsilon$. The expression can be simplified as $J = \tilde{\mathcal{O}}\left(\frac{\log(\frac{1}{\delta})}{\epsilon^2(1-\gamma)^6}\right)$ and $L = \tilde{\mathcal{O}}\left(\frac{1}{1-\gamma}\right)$, omitting the logarithmic and constant terms. That leads to

$$N = \tilde{\mathcal{O}}\left(\frac{\log(\frac{1}{\delta})}{\epsilon^2(1-\gamma)^7}\right). \tag{33}$$

C PROOF OF LEMMAS

In this section, we provide the proof of auxiliary lemmas.

C.1 Proof of Lemma 3 [RKHS norm of PV]

We have

$$\begin{aligned}\|PV\|_{\mathcal{H}_K} &= \left\|\int_{s'\in\mathcal{S}} P(s'|\cdot,\cdot)V(s')ds'\right\|_{\mathcal{H}_K}\\ &\leq \int_{s'\in\mathcal{S}} \|P(s'|\cdot,\cdot)V(s')\|_{\mathcal{H}_K} ds'\\ &= \int_{s'\in\mathcal{S}} \|P(s'|\cdot,\cdot)\|_{\mathcal{H}_K} V(s')ds'\\ &\leq \int_{s'\in\mathcal{S}} V(s')ds'. \end{aligned} \tag{34}$$

where the last inequality holds by Assumption 2. We note that $\int_{s'\in\mathcal{S}} V(s')ds' \leq \frac{1}{1-\gamma}\int_{s'} ds' \leq \frac{c}{1-\gamma}$ where c is the volume of $\mathcal{S}$.

C.2 Proof of Lemma 4 [Sub-Gaussianity of PV]

The first part, $\mathbb{E}\left[[\widehat{P}V](s,a)\right] = PV(s,a)$, follows from the definition of $\widehat{P}V$. For the second part note that $\widehat{P}V$ is a random variable with a bounded support in $[0, \frac{1}{1-\gamma}]$ by definition. Hoeffding lemma states that: let X be any random variable such that $0 \leq X \leq a$, then for any $\zeta \in \mathbb{R}$, $\mathbb{E}\left[\exp\left(\zeta(X-\mathbb{E}[X])\right)\right] \leq \exp(\frac{\zeta^2 a^2}{8})$. Applying Hoeffding lemma, we can see that $\widehat{P}V$ is sub-Gaussian with parameter $\frac{1}{2(1-\gamma)}$.

C.3 Proof of Lemma 5 [Error in value function based on the error in proxy Q-function]

The proof follows similar steps as the proof of Lemma 1.11 in Agarwal et al. (2019) which considered finite state-actions. First, recall the following definitions from Section 2,

$$Q^{\pi_{\tilde{Q}}}(s,a) = r(s,a) + \gamma[PV^{\pi_{\tilde{Q}}}](s,a) \quad \text{and} \quad Q^*(s,a) = \max_{\pi} Q(s,\pi(s)).$$

Fix state $s \in \mathcal{S}$ and let $\tilde{a}_s = \pi_{\tilde{Q}}(s)$. We have

$$\begin{aligned} V^*(s) - V^{\pi_{\tilde{Q}}}(s) =& Q^*\left(s,\pi^*(s)\right) - Q^{\pi_{\tilde{Q}}}(s,\tilde{a}_s)\\ =& Q^*\left(s,\pi^*(s)\right) - Q^*(s,\tilde{a}_s) + Q^*(s,\tilde{a}_s) - Q^{\pi_{\tilde{Q}}}(s,\tilde{a}_s)\\ =& Q^*\left(s,\pi^*(s)\right) - Q^*(s,\tilde{a}_s) + \gamma[PV^*](s,\tilde{a}_s) - \gamma[PV^{\pi_{\tilde{Q}}}](s,\tilde{a}_s)\\ =& Q^*\left(s,\pi^*(s)\right) - Q^*(s,\tilde{a}_s) + \gamma\mathbb{E}_{s'\sim P(\cdot|s,\tilde{a}_s)}\left[V^*\left(s'\right) - V^{\pi_{\tilde{Q}}}\left(s'\right)\right]\\ \leq& Q^*\left(s,\pi^*(s)\right) - \tilde{Q}\left(s,\pi^*(s)\right) + \tilde{Q}(s,\tilde{a}_s) - Q^*(s,\tilde{a}_s)\\ &+ \gamma\mathbb{E}_{s'\sim P(s,\tilde{a}_s)}\left[V^*\left(s'\right) - V^{\pi_{\tilde{Q}}}\left(s'\right)\right]\\ \leq& 2\left\|\tilde{Q} - Q^*\right\|_\infty + \gamma\left\|V^* - V^{\pi_{\tilde{Q}}}\right\|_\infty, \end{aligned}$$

where the first inequality uses $\tilde{Q}\left(s,\pi^*(s)\right) \leq \tilde{Q}(s,\tilde{a}_s)$ by definition of $\tilde{a}_s$. We thus have

$$\left\|V^* - V^{\pi_{\tilde{Q}}}\right\|_\infty \leq 2\left\|\tilde{Q} - Q^*\right\|_\infty + \gamma\left\|V^* - V^{\pi_{\tilde{Q}}}\right\|_\infty. \tag{35}$$

Rearranging this inequality, we arrive at the lemma

$$\left\|V^* - V^{\pi_{\tilde{Q}}}\right\|_\infty \leq \frac{2}{1-\gamma}\left\|\tilde{Q} - Q^*\right\|_\infty. \tag{36}$$