Reward-Free Kernel-Based Reinforcement Learning

Sattar Vakili [1] **Farhang Nabiei** [1] **Da-shan Shiu** [1] **Alberto Bernacchia** [1]

Abstract

Achieving sample efficiency in Reinforcement Learning (RL) is primarily hinged on the efficient exploration of the underlying environment, but it is still unknown what are the best exploration strategies in different settings. We consider the *reward-free* RL problem, which operates in two phases: an exploration phase, where the agent gathers exploration trajectories over episodes irrespective of any predetermined reward function, and a subsequent planning phase, where a reward function is introduced. The agent then utilizes the episodes from the exploration phase to calculate a near-optimal policy. Existing algorithms and sample complexities for reward-free RL are limited to tabular, linear or very smooth function approximations, leaving the problem largely open for more general cases. We consider a broad range of kernel-based function approximations, including non-smooth kernels, and propose an algorithm based on adaptive domain partitioning. We show that our algorithm achieves order-optimal sample complexity for a large class of common kernels, which includes Matérn and Neural Tangent kernels.

1. Introduction

Reinforcement Learning (RL) policies using complex function approximations have been empirically effective in various 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). To achieve sample efficiency, these RL policies must learn the transition model, either directly or indirectly, necessitating efficient exploration. In the context of offline RL, the agent learns the optimal policy solely from a pre-collected offline dataset, without any further interaction with the environment. Therefore, the offline dataset should adequately represent the trajectory produced by the optimal policy. In real-world RL applications, the reward function is often crafted by the learner based on domain knowledge. The learner may have multiple reward functions to select from or may employ an adaptive algorithm for reward design. In such situations, it is preferable to have an offline dataset that encapsulates all potential optimal trajectories associated with a variety of reward functions. With such a comprehensive offline dataset, the RL agent can estimate the corresponding policy for any arbitrary reward function.

To methodically study this problem, we concentrate on the reward-free RL framework, which includes an *exploration* phase and a *planning* phase (Figure 1). In the exploration phase, the agent interacts with the environment without any pre-determined rewards and gathers empirical trajectories over episodes for the subsequent planning phase. During the planning phase, the agent uses the offline data accumulated in the exploration phase to compute the optimal policy for a given extrinsic reward function r, without further interactions with the environment.

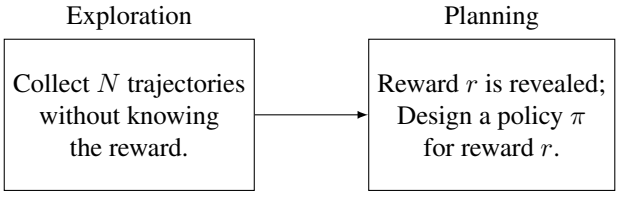

Figure 1. Reward-Free RL framework.

The reward-free RL framework has been progressively examined under increasingly complex models —*tabular* → *linear* → *kernel-based* → *deep learning based*— in several works including (Jin et al., 2020a; Wang et al., 2020; Qiu et al., 2021). The existing literature adequately addresses the tabular and linear settings. It however tends to falter, providing only partial and incomplete results when dealing with the more intricate kernel-based and deep learning based settings. The contribution of this paper is to further the literature by providing order optimal results in the kernel-based setting, applicable to a broad class of common kernels.

Our main objectives are: *(i)* Designing algorithms for both

[1]MediaTek Research. Correspondence to: Sattar Vakili <sattar.vakili@mtkresearch.com>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

exploration and planning phases in the reward-free RL framework with kernel-based modeling. *(ii)* Improving the existing results by proving order optimal sample complexities for a broad class of common kernels.

To touch on the technical importance of our results, we show in Table 1 the *sample complexity* in various settings. The sample complexity refers to the number of exploration episodes collected in the exploration phase to obtain ϵ-optimal policies in the planning phase, where a policy is referred to as ϵ-optimal if its value function is at most ϵ away from the optimal value function (See Definition 2.1). Here, $\mathcal{S}$, $\mathcal{A}$ and $H \in \mathbb{N}$ represent the state and action spaces and the episode length, respectively. In the kernel-based setting, previous work (Qiu et al., 2021) provided algorithms whose sample complexities are, up to logarithmic factors, order optimal with respect to ϵ. However, these results are primarily applicable to very smooth kernels, specifically those characterized by exponentially decaying eigenvalues. This limitation effectively excludes significant kernel families, such as Matérn and Neural Tangent kernels.

For a more nuanced understanding of the existing results, let $\{\lambda_m > 0\}_{m=1}^{\infty}$ represent the Mercer eigenvalues of kernel k, sequenced in diminishing order, and let $\{\phi_m\}_{m=1}^{\infty}$ denote the corresponding eigenfeatures. For details, refer to Section 2.3. The kernel k is characterized as having an exponential eigendecay when its eigenvalues λ_m diminish exponentially with respect to m, specifically $\lambda_m = \mathcal{O}(\iota^m)$ for some $0 < \iota < 1$; an example being the Squared Exponential kernel. In contrast, the kernel k is described as having a polynomial eigendecay when its eigenvalues λ_m decline at a rate no slower than m^{-p} for some $p > 1$. This decay profile is characteristic of numerous kernels, both of practical importance and theoretical interest, such as the Matérn family of kernels (Borovitskiy et al., 2020) and the Neural Tangent (NT) kernel (Arora et al., 2019). Specifically, for a Matérn kernel with smoothness parameter ν in a d-dimensional space, $p = \frac{2\nu+d}{d}$ (e.g., see, Yang et al., 2020a). Similarly, for an NT kernel with $s-1$ times differentiable activations, $p = \frac{2s-1+d}{d}$ (Vakili et al., 2021b).

Leveraging the scaling of the kernel spectrum with the size of the domain can improve the sample complexity. We focus on kernels with polynomial eigendecay within a hypercubical domain of side length ρ, where eigenvalues scale as $m^{-p}\rho^{\alpha}$ for some $\alpha > 0$, as detailed in Definition 4.1. This approach covers a broad spectrum of prevalent kernels, such as the Matérn family, where $\alpha = 2\nu$. While we employ a hypercube domain for technical consistency, this assumption is flexible and can extend to other regular, compact subsets of $\mathbb{R}^d$.

Our contribution lies in devising algorithms for both the exploration and planning phases of the reward-free RL framework, establishing sample complexities for kernels with polynomially decaying eigenvalues. We demonstrate that an $\tilde{\mathcal{O}}\left((\frac{H^3}{\epsilon})^{2+\frac{2d}{\alpha}}\right)$ exploration episodes are sufficient to guarantee ϵ-optimal policies during planning. When applied to the Matérn kernel, our sample complexity becomes $\tilde{\mathcal{O}}\left((\frac{H^3}{\epsilon})^{2+\frac{d}{\nu}}\right)$. This is a significant improvement, contrasted with existing work (Qiu et al., 2021), where the sample complexity becomes unbounded for many parameter values of the Matérn kernel, such as when $\nu < \frac{d(d+1)}{2}$. In addition, our sample complexities match the $\Omega\left((\frac{1}{\epsilon})^{2+\frac{d}{\nu}}\right)$ lower bound for the kernel-based bandit problem with Matérn kernel (see, Scarlett et al., 2017, *Table I*) that can be considered as a degenerate special case with $H = 1$, $|\mathcal{S}| = 1$, indicating that the performance in ϵ cannot be further improved.

We obtain these samples complexities by designing algorithms tailored for the polynomial class of kernels. The main design ideas include leveraging a *hypothetical reward* for the exploration phase proportional to the uncertainty in kernel-based regression, and an adaptive domain partitioning technique inspired by the recent work of Vakili & Olkhovskaya (2023). In this method, the algorithm creates a partitioning of the state-action domain and builds value function estimates only based on the observations within the same partition element. See details in Section 3.

In Section 2, we present the episodic Markov Decision

Table 1. Existing results on the sample complexity of reward-free RL. The sample complexity refers to the number of exploration episodes collected in the exploration phase to obtain ϵ-optimal policies in the planning phase. The notation $\mathcal{S}$, $\mathcal{A}$ and H denote the state and action spaces and the episode length respectively. The parameter d denotes the state action space dimension and α represents smoothness properties of the kernel. While the existing results fail to obtain even finite sample complexities in the general kernel-based setting, we report order optimal sample complexity, given in the last row of this table.

SETTING	SAMPLE COMPLEXITY
TABULAR (MÉNARD ET AL., 2021)	$\mathcal{O}\left(\frac{\lvert\mathcal{S}\rvert^2\lvert\mathcal{A}\rvert H^3}{\epsilon^2}\right)$
LINEAR (WAGENMAKER ET AL., 2022)	$\tilde{\mathcal{O}}\left(\frac{d^2H^5}{\epsilon^2}\right)$
KERNEL-BASED WITH EXPONENTIAL EIGENDECAY (QIU ET AL., 2021)	$\mathcal{O}\left(\frac{H^6 polylog(\frac{1}{\epsilon})}{\epsilon^2}\right)$
KERNEL-BASED WITH POLYNOMIAL EIGENDECAY (THIS WORK)	$\boldsymbol{\tilde{\mathcal{O}}\left((\frac{H^3}{\epsilon})^{2+\frac{2d}{\alpha}}\right)}$

Process (MDP) setting, formalize reward-free RL framework and our assumptions, and overview the kernel ridge regression. In Section 4, we present the sample complexity analysis.

1.1. Literature Review

The reward-free RL framework under the episodic setting has been studied with tabular model in Jin et al. (2020a); Zhang et al. (2020); Ménard et al. (2021); Kaufmann et al. (2021), and with linear model in Wang et al. (2020); Zanette et al. (2020c); Wagenmaker et al. (2022), with the best sample complexities reported in Table 1. The problem has also been studied under the linear mixture model in Zhang et al. (2021); Chen et al. (2021); Zhang et al. (2023).

The sample complexity of the RL problem on a discounted MDP setting with an infinite horizon has been considered under various tabular, linear and kernel-based settings in (Kearns & Singh, 1998; Azar et al., 2013; Sidford et al., 2018; Agarwal et al., 2020; Yang & Wang, 2019; Yeh et al., 2023). These works however assume the existence of a generative *oracle* (Kakade, 2003), which provides sample transitions from any state-action pair of algorithm's choice. This assumption simplifies the problem compared to the reward-free RL framework considered in this work, where the agent must follow the MDP trajectory within each episode and cannot arbitrarily inquire transitions from state-action pairs. Specifically, we design an exploration algorithm based on uncertainty estimates obtained from the kernel-based model that adds significant challenges to the analysis and is not required in the oracle setting.

Our algorithm design is inspired by the domain partitioning technique used in Vakili & Olkhovskaya (2023), as well as in Janz et al. (2020) for kernel-based bandits. In comparison, Vakili & Olkhovskaya (2023) considered the standard episodic RL setting, where the reward function is known to the policy a priori. That is different from the reward-free RL framework considered in this work and their results do not apply here.

There is an extensive literature on the analysis of RL policies which do not rely on a generative model or an exploratory behavioral policy. The literature has primarily focused on the tabular setting (Jin et al., 2018; Auer et al., 2008; Bartlett & Tewari, 2012). Recent literature has placed a notable emphasis on employing function approximation in RL, particularly within the context of generalized linear settings. This approach involves representing the value function or transition model through a linear transformation to a well-defined feature mapping. Important contributions include the work of Jin et al. (2020b); Yao et al. (2014), as well as subsequent studies by Russo (2019); Zanette et al. (2020a;b); Neu & Pike-Burke (2020); Yang & Wang (2020). Furthermore, there have been several efforts to extend these techniques to a kernelized setting, as explored in Yang et al. (2020a); Yang & Wang (2020); Chowdhury & Gopalan (2019); Yang et al. (2020b); Domingues et al. (2021). These works are also inspired by methods originally designed for linear bandits (Abbasi-Yadkori et al., 2011; Agrawal & Goyal, 2013), as well as kernelized bandits (Srinivas et al., 2010; Valko et al., 2013; Chowdhury & Gopalan, 2017).

2. Problem Formulation

In this section, we present the episodic MDP setting, the reward-free RL framework, background on kernel methods and our technical assumptions.

2.1. Episodic MDP

An episodic MDP can be described by the tuple $M = (\mathcal{S}, \mathcal{A}, H, P, r)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, the integer H is the length of each episode, $r = \{r_h\}_{h=1}^H$ are the reward functions and $P = \{P_h\}_{h=1}^H$ are the transition probability distributions.[1] We use the notation $\mathcal{Z} = \mathcal{S} \times \mathcal{A}$ to denote the state-action space. For each $h \in [H]$, the reward $r_h : \mathcal{Z} \to [0, 1]$ is the reward function at step h, which is supposed to be deterministic for simplicity, and $P_h(\cdot|s, a)$ is the unknown transition probability distribution on $\mathcal{S}$ for the next state from state-action pair (s, a).

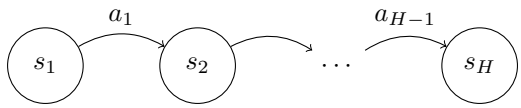

Figure 2. Illustration of an Episodic MDP with episode of length H.

A policy $\pi = \{\pi_h\}_{h=1}^H$, at each step h, determines the (possibly random) action $\pi_h : \mathcal{S} \to \mathcal{A}$ taken by the agent at state s. At the beginning of each episode, the environment picks an arbitrary state s_1. The agent determines a policy $\pi = \{\pi_h\}_{h=1}^H$. Then, at each step $h \in [H]$, the agent observes the state $s_h \in \mathcal{S}$, and picks an action $a_h = \pi_h(s_h)$. The new state s_{h+1} then is drawn from the transition distribution $P_h(\cdot|s_h, a_h)$. The episode ends when the agent receives the final reward $r_H(s_H, a_H)$.

We are interested in maximizing the expected total reward in the episode, starting at step h, i.e., the value function, defined as

$$V_h^\pi(s) = \mathbb{E}\left[\sum_{h'=h}^{H} r_{h'}(s_{h'}, a_{h'}) \middle| s_h = s\right], \quad \forall s \in \mathcal{S}, h \in [H], \tag{1}$$

[1] We intentionally do not use the standard term transition kernel for P_h, to avoid confusion with the term kernel in kernel-based learning.

where the expectation is taken with respect to the randomness in the trajectory $\{(s_h, a_h)\}_{h=1}^H$ obtained by the policy π. It can be shown that under mild assumptions (e.g., continuity of P_h, compactness of $\mathcal{Z}$, and boundedness of r) there exists an optimal policy $\pi^\star$ which attains the maximum possible value of $V_h^\pi(s)$ at every step and at every state (e.g., see, Puterman, 2014). We use the notation $V_h^\star(s) = \max_\pi V_h^\pi(s)$, $\forall s \in \mathcal{S}, h \in [H]$. By definition $V_h^{\pi^\star} = V_h^\star$. An ϵ-optimal policy is defined as follows.

Definition 2.1. (ϵ-optimal policy) For $\epsilon > 0$, A policy π is called ϵ-optimal if it achieves near-optimal values from any initial state as follows:

$$V_1^\pi(s) \geq V_1^\star(s) - \epsilon, \quad \forall s \in \mathcal{S}.$$

For a value function V, we define the following notation

$$[P_h V](s, a) := \mathbb{E}_{s' \sim P_h(\cdot|s,a)}[V(s')]. \tag{2}$$

We also define the state-action value function $Q_h^\pi : \mathcal{Z} \to [0, H]$ as follows.

$$Q_h^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{h'=h}^{H} r_{h'}(s_{h'}, a_{h'}) \middle| s_h = s, a_h = a \right], \tag{3}$$

where the expectation is taken with respect to the randomness in the trajectory $\{(s_h, a_h)\}_{h=1}^H$ obtained by the policy π. The Bellman equation associated with a policy π then is represented as

$$\begin{aligned} Q_h^\pi(s, a) &= r_h(s, a) + [P_h V_{h+1}^\pi](s, a), \\ V_h^\pi(s) &= \max_a Q_h^\pi(s, a), \quad V_{H+1}^\pi = \mathbf{0}. \end{aligned}$$

where the expectation is taken with respect to the randomness in the policy π. We may specify the reward function in $V^\pi, Q^\pi, V^\star, Q^\star$ notations for clarity, for example, $V^\pi(s; r)$ and $Q^\star(z; r)$.

2.2. Reward-Free RL Framework

We aim to learn ϵ-optimal policies using as small as possible number of collected exploration episodes. In particular, we consider the reward-free RL framework that consists of two phases: exploration and planning. In the exploration phase, we collect N exploration episodes $\{(s_1^n, a_1^n, s_2^n, a_2^n, \cdots, s_H^n)\}_{n=1}^N$ without any revealed reward function. Then, in the planning phase, reward r is revealed, and we design a policy for reward r using the trajectories collected in the exploration phase. We refer to N as the sample complexity of designing ϵ-optimal policy. Under certain assumptions, the question is: *How many exploration episodes are required to obtain ϵ-optimal policies?* We provide an answer in Theorem 4.5.

2.3. Kernel Ridge Regression

We assume that the unknown transition probability distribution can be represented using a reproducing kernel Hilbert space (RKHS). See Assumption 2.2. This is a very general assumption, considering that the RKHS of common kernels can approximate almost all continuous functions on the compact subsets of $\mathbb{R}^d$ (Srinivas et al., 2010). Consider a positive definite kernel $k : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$. 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:

$$k(z, z') = \sum_{m=1}^{\infty} \lambda_m \phi_m(z) \phi_m(z'), \tag{4}$$

where $\lambda_m > 0$, and $\sqrt{\lambda_m}\phi_m \in \mathcal{H}_k$ form an orthonormal basis of $\mathcal{H}_k$. In particular, any $f \in \mathcal{H}_k$ can be represented using this basis and wights $w_m \in \mathbb{R}$ as

$$f = \sum_{m=1}^{\infty} w_m \sqrt{\lambda_m} \phi_m, \tag{5}$$

where $\|f\|_{\mathcal{H}_k}^2 = \sum_{m=1}^{\infty} w_m^2$. A formal statement and the details are provided in Appendix A. We refer to λ_m and ϕ_m as (Mercer) eigenvalues and eigenfeatures of k, respectively.

Kernel-based models provide powerful predictors and uncertainty estimators, which can be leveraged to guide the RL algorithm. In particular, consider a fixed unknown function $f \in \mathcal{H}_k$. Consider a set $Z^n = \{z^i\}_{i=1}^n \subset \mathcal{Z}$ of n inputs. Assume n noisy observations $\{Y(z^i) = f(z^i) + \varepsilon^i\}_{i=1}^n$ are provided, where ε^i are independent zero mean noise terms. Kernel ridge regression provides the following predictor and uncertainty estimate, respectively (see, e.g., Schölkopf et al., 2002),

$$\begin{aligned} \mu^{n,f}(z) &= k_{Z^n}^\top(z)(K_{Z^n} + \tau^2 I^n)^{-1} Y_{Z^n}, \\ (\sigma^n(z))^2 &= k(z, z) - k_{Z^n}^\top(z)(K_{Z^n} + \tau^2 I^n)^{-1} k_{Z^n}(z), \end{aligned} \tag{6}$$

where $k_{Z^n}(z) = [k(z, z^1), \ldots, k(z, z^n)]^\top$ is a $n \times 1$ vector of the kernel values between z and observations, $K_{Z^n} = [k(z^i, z^j)]_{i,j=1}^n$ is the $n \times n$ kernel matrix, $Y_{Z^n} = [Y(z^1), \ldots, Y(Z^n)]^\top$ is the $n \times 1$ observation vector, I^n is the identity matrix of dimensions n, and $\tau > 0$ is a free regularization parameter. The predictor and uncertainty estimate could be interpreted as posterior mean and variance of a surrogate centered Gaussian process (GP) model with covariance k, and zero mean Gaussian noise with variance τ^2 (e.g., see, Williams & Rasmussen, 2006).

2.4. Technical Assumption

We assume that the unknwon transition probability distributions $P_h(s'|\cdot,\cdot)$ of the MDP belong to the 1-ball of the RKHS. We use the notation $\mathcal{B}_{k,R} = \{f : \|f\|_{\mathcal{H}_k} \leq R\}$ to denote the R-ball of the RKHS.

Assumption 2.2. We assume

$$P_h(s'|\cdot,\cdot) \in \mathcal{B}_{k,1}, \quad \forall h \in [H],\ \forall s' \in \mathcal{S}. \tag{7}$$

This is a mild assumption considering the generality of RKHSs, that is also supposed to hold in (Qiu et al., 2021; Yang et al., 2020a). Similar assumptions are made in linear MDPs which are significantly more restrictive (e.g., see, Wang et al., 2020; Jin et al., 2020b).

A consequence of Assumption 2.2 is that for any integrable $V : [0,1]^d \to [0,H]$, $[P_h V_{h+1}] \in \mathcal{B}_{k,H}$. This is formalized in the following lemma (See, Yeh et al., 2023, Lemma 3).

Lemma 2.3. *Consider any integrable $V : [0,1]^d \to [0,H]$. Under Assumption 2.2, we have*

$$[P_h V] \in \mathcal{B}_{k,H}. \tag{8}$$

3. Algorithm

In this section, we design novel algorithms for both exploration and planning phases in the kernel-based reward-free RL framework described in Section 2.

The two main ideas in our design are *(i)* the use of a *hypothetical* reward in the exploration phase and *(ii) domain partitioning* in application of kernel-based confidence intervals.

Hypothetical reward. In the exploration phase, we will craft a carefully chosen hypothetical reward function that incentivizes efficient exploration. We choose the term hypothetical reward since it is different from the actual rewards revealed to the agent later in the planning phase. In other words, in the exploration phase when the reward is yet not revealed to the agent, we design a notion of reward for the agent that encourages an efficient exploration. We use the uncertainty estimates provided by kernel regression to define the hypothetical reward and motivate exploration of uncertain regions in the state-action space. In particular, for episode n in the exploration phase, we choose hypothetical reward $\tilde{r}^n = \beta(\delta)\sigma^n/H$, where σ_h^n is the posterior standard deviation of the kernel-based model conditioned on (possibly some of) past $n-1$ previous episodes, and $\beta(\delta)$ is a $1-\delta$ confidence interval multiplier that is specified in Theorem 4.5. Using this hypothetical reward incentivizes the agent to move on the Markovian trajectory towards state-actions with higher uncertainty.

This is different from a pure and uniform exploration. This is also different from (Yeh et al., 2023), where the existence of a generative oracle was assumed that can provide transition samples for the state-action pairs of the agent's choice. The reward-free RL framework considered in this work is more sophisticated in the sense that the agent must stay on the Markovian trajectory and cannot observe arbitrary state-actions.

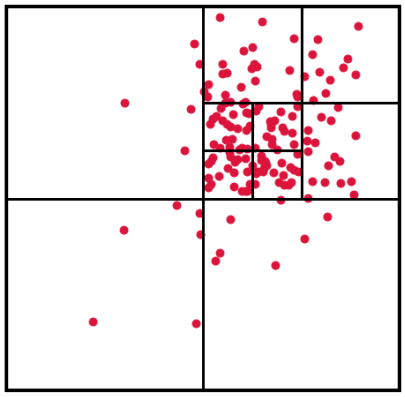

Figure 3. An example of adaptive domain partitioning on $\mathcal{Z} = [0,1]^2$. The dots represent a sequence of points. Partitions are created by dividing every square $\mathcal{Z}'$ that satisfies the condition $\rho_{\mathcal{Z}'}^{-\alpha} < N(\mathcal{Z}') + 1$ into four equal smaller squares. Here, $\rho_{\mathcal{Z}'}$ and $N(\mathcal{Z}')$ denote the side length of a square $\mathcal{Z}'$ and the number of dots within $\mathcal{Z}'$, respectively. In this example, α is set to 3.

Domain partitioning. In both the exploration and planning phases, we will use domain partitioning to improve the precision of prediction and analytical guarantees on the approximations. In particular, we partition the state-action space into subdomains and only use the observations within the same subdomain for kernel-based prediction (disregarding the rest of the observations). As shown in Vakili & Olkhovskaya (2023), this allows a tradeoff between the standard deviation of the kernel-based model and the confidence interval width coefficient. An optimal procedure for domain partitioning leads to an improved performance. An example of partitioning on a 2-dimensional state-action domain is shown in Figure 3.

3.1. Exploration Phase

The exploration algorithm simply employs an optimistic least-squares value iteration (LSVI) with the hypothetical reward. Optimistic LSVI is a standard policy in episodic MDPs, which, inspired by the principle of *optimism in the face of uncertainty*, computes an upper confidence bound on state-action value function. For this purpose, kernel ridge regression is used to form prediction f_h^n and uncertainty estimate σ_h^n for the $[P_h V_{h+1}]$ term in the state-action value function. Then the upper confidence bound is defined as

$$Q_h^n(\cdot,\cdot) = \Pi_{[0,H]}\left[\tilde{r}_h^n(\cdot,\cdot) + f_h^n(\cdot,\cdot) + \beta(\delta)\sigma_h^n(\cdot,\cdot)\right]. \tag{9}$$

Algorithm 1 Exploration Phase

Input: τ, $\beta(\delta)$, k, $\mathcal{S}, \mathcal{A}, H, P$.
For all $h \in [H]$, let $\mathcal{S}_h^1 = \{[0,1]^d\}$.
for Episode $n = 1, 2, \ldots, N$, **do**
  Receive the initial state s_1^n.
  Set $V_{H+1}^n(s) = 0$, for all s.
  for step $h = H, \ldots, 1$ **do**
    Obtain Q_h^n as in (9) based on (11) and (12).
    $V_h^n(\cdot) = \max_{a \in \mathcal{A}} Q_h(\cdot, a)$.
  end for
  for step $h = 1, 2, \ldots, H$ **do**
    Take action $a_h^n \leftarrow \text{argmax}_{a \in \mathcal{A}} Q_h^n(s_h^n, a)$.
    Receive the next state s_{h+1}^n.
    Split any element $\mathcal{Z}' \in \mathcal{S}_h^{n-1}$, for which $\rho_{\mathcal{Z}'}^{-\alpha} < |N_h^n(\mathcal{Z}')| + 1$ along the middle of each side, and obtain $\mathcal{S}_h^n$.
  end for
end for

The notation $\Pi_{[a,b]}[\cdot]$ denotes projection onto interval $[a, b]$. Since the rewards are assumed to be at most 1, the state-action value function at step h is also bounded by H, hence projection to $[0, H]$ interval. In episode n, then π^n is the greedy policy with respect to $Q^n = \{Q_h^n\}_{h=1}^H$. Under Assumption 2.2, the estimate f_h^n, the parameter $\beta(\delta)$ and the uncertainty estimate σ_h^n can all be designed using kernel ridge regression.

To overcome the suboptimal performance guarantees rooted in the online confidence intervals in kernel ridge regression, we use a carefully designed domain partitioning. The proposed algorithm partitions the state-action space $\mathcal{Z}$ into subdomains and builds kernel ridge regression only based on the observations within each subdomain. By doing so, we obtain tighter confidence intervals, ultimately resulting in tighter performance guarantees.

To formalize this procedure, we consider the state-action space $\mathcal{Z} \subset [0,1]^d$. Let $\mathcal{S}_h^n$, $h \in [H], n \in [N]$ be sets of hypercubes overlapping only at edges, covering the entire $[0,1]^d$. For any hypercube $\mathcal{Z}' \in \mathcal{S}_h^n$, we use $\rho_{\mathcal{Z}'}$ to denote the length of any of its sides, and $N_h^n(\mathcal{Z}')$ to denote the number of observations at step h in $\mathcal{Z}'$ up to episode n:

$$N_h^n(\mathcal{Z}') = \sum_{i=1}^{n} \mathbf{1}\{(s_h^i, a_h^i) \in \mathcal{Z}'\}. \tag{10}$$

For all $h \in [H]$, we initialize $\mathcal{S}_h^1 = \{[0,1]^d\}$. At exploration episode n, for each step h, after observing a sample from $[P_h V_{h+1}^n]$ at (s_h^n, a_h^n), we construct a new cover $\mathcal{S}_h^n$ as follows. We divide every element $\mathcal{Z}' \in \mathcal{S}_h^{n-1}$ that satisfies $\rho_{\mathcal{Z}'}^{-\alpha} < N_h^n(\mathcal{Z}') + 1$, into two equal halves along each side, generating 2^d hypercubes. The parameter $\alpha > 0$ in the splitting rule is a constant specified in Definition 4.1. Subsequently, we define $\mathcal{S}_h^n$ as the set of newly created hypercubes and the elements of $\mathcal{S}_h^{n-1}$ that were not split.

The construction of the cover sets described above ensures the number $N_h^n(\mathcal{Z}')$ of observations within each cover element $\mathcal{Z}'$ remains relatively small taking into account the size of $\mathcal{Z}'$, while also controlling the total number $|\mathcal{S}_h^n|$ of cover elements. The key parameter managing this tradeoff is α, which is carefully chosen to achieve an appropriate width for the confidence interval, as shown in Section 4.

Our exploration algorithm is derived by adopting the structure of the optimistic LSVI, as described above, where the predictor and the uncertainty estimates are designed based on kernel ridge regression only on cover elements. In particular, for $z \in \mathcal{Z}$, let $\mathcal{Z}_h^n(z) \in \mathcal{S}_h^n$ be the cover element at step h of episode n containing z. Define $Z_h^n(z) = \{(s_h^i, a_h^i) \in \mathcal{Z}_h^n(z), i < n\}$ to be the set of past observations belonging to the same cover element as z. We then set

$$f_h^n(z) = k_{Z_h^n(z)}^\top(z)(K_{Z_h^n(z)} + \tau^2 I)^{-1} Y_{Z_h^n(z)}, \tag{11}$$

where $k_{Z_h^n(z)} = [k(z, z')]_{z' \in Z_h^n(z)}^\top$ is the kernel values between z and all observations z' in $Z_h^n(z)$, $K_{Z_h^n(z)} = [k(z', z'')]_{z', z'' \in Z_h^n(z)}$ is the kernel matrix for observations in $Z_h^n(z)$. Starting from $V_{H+1} = \mathbf{0}$, for $h = H, \cdots, 1$, we obtain the observation value as follows: $Y_{Z_h^n(z)} = [V_{h+1}^n(s'_{h+1})]_{z' \in Z_h^n(z)}^\top$, where s'_{h+1} is drawn from the transition distribution $P_h(\cdot|z')$, denotes the observation values for the observation points $z' \in Z_h^n(z)$. The vectors $k_{Z_h^n(z)}$ and $Y_{Z_h^n(z)}$ are $N_h^{n-1}(\mathcal{Z}_h^n(z))$ dimensional column vectors, and $K_{Z_h^n(z)}$ and I are $N_h^{n-1}(\mathcal{Z}_h^n(z)) \times N_h^{n-1}(\mathcal{Z}_h^n(z))$ dimensional matrices.

Note that, having the Bellman equation in mind, f_h^n is the (kernel ridge) predictor for $[P_h V_{h+1}^n]$ using some of the past $n-1$ observations $\{V_{h+1}^n(s_{h+1}^i)\}_{i=1}^{n-1}$ at points $\{z_h^i\}_{i=1}^{n-1}$. Recall that $\mathbb{E}\left[V_{h+1}^n(s_{h+1}^i)\right] = [P_h V_{h+1}^n](z_h^i)$, where the expectation is taken with respect to $P_h(\cdot|z_h^i)$. The observation noise $V_{h+1}^n(s_{h+1}^i) - [P_h V_{h+1}^n](z_h^i)$ is due to random transitions and is bounded by $H - h \le H$.

The exploration bonus is determined based on the uncertainty estimate of the kernel ridge regression model on cover elements defined as

$$\sigma_h^n(z) = \sqrt{k(z,z) - k_{Z_h^n(z)}^\top(z)(K_{Z_h^n(z)} + \tau^2 I)^{-1} k_{Z_h^n(z)}(z)}. \tag{12}$$

The policy then is the greedy policy with respect to Q_h^n given in (9). Specifically, at step h of exploration episode n, the following action is chosen, after observing s_h^n,

$$a_h^n = \text{argmax}_{a \in \mathcal{A}} Q_h^n(s_h^n, a). \tag{13}$$

A pseudocode is provided in Algorithm 1.

3.2. Planning Phase

In the planning phase, when the reward function r is revealed, a near-optimal policy π is derived using the episodes of trajectories collected during the exploration phase. Similar to the exploration policy, we compute a prediction g_h and a confidence interval width w_h for $[P_h V_{h+1}]$, and define

$$Q_h(\cdot,\cdot) = \Pi_{[0,H]}[r_h(\cdot,\cdot) + g_h(\cdot,\cdot) + w_h^n(\cdot,\cdot)]. \quad (14)$$

The policy π then is obtained as a greedy policy with respect to Q

$$\pi_h(\cdot) = \text{argmax}_{a\in\mathcal{A}} Q_h(\cdot, a).$$

Due to domain partitioning, the confidence interval width may increase over the exploration episode n for certain points. To address this specific observation related to the domain partitioning technique, we take the following steps. First, we identify the exploration episode that has the smallest confidence interval for $z \in \mathcal{Z}$. Specifically, let us define

$$w_h(z) = \min_{n\le N} \beta(\delta)\sigma_h^n(z). \quad (15)$$

Also, let $n_h(z) = \arg\min_{n\le N} \beta(\delta)\sigma_h^n(z)$ be the exploration episode that provides the tightest confidence interval for point z. Recall that for $z \in \mathcal{Z}$, we defined $\mathcal{Z}_h^n(z) \in \mathcal{S}_h^n$ to be the cover element at step h of episode n containing z, and $Z_h^n(z) = \{(s_h^i, a_h^i) \in \mathcal{Z}_h^n(z), i < n\}$ as the set of past observations belonging to the same cover element as z.

Keeping Bellman equation in mind, and starting with $V_{H+1} = \mathbf{0}$, g_h is the kernel ridge predictor for $[P_h V_{h+1}]$ using some of the n observations $\{V_{h+1}(s_{h+1}^n)\}_{n=1}^N$ at points $\{z^n\}_{n=1}^N$ in the exploration phase. Specifically, in computing $g_h(z)$, we only use observations that are within the same subdomain as z in the partition $\mathcal{S}_h^{n_h(z)}$ at episode $n_h(z)$ of exploration phase

$$g_h(z) = k_{Z_h^{n_h(z)}(z)}^\top(z)(K_{Z_h^{n_h(z)}(z)} + \tau^2 I)^{-1}\tilde{Y}_{Z_h^{n_h(z)}(z)}, \quad (16)$$

where $\tilde{Y}_{Z_h^{n_h(z)}(z)} = [V_{h+1}(s'_{h+1})]^\top_{z'\in Z_h^{n_h(z)}(z)}$. A pseudocode is given in Algorithm 2.

Algorithm 2 Planning Phase

Input: τ, $\beta(\delta)$, k, $M(\mathcal{S},\mathcal{A},H,P,r)$, and exploration data $\{(s_h^n, a_h^n)\}_{(h,n)\in[H]\times[N]}$
for $h = H, H-1, \cdots, 1$, **do**
  Compute the prediction g_h
  Let $Q_h(\cdot,\cdot) = \Pi_{[0,H]}[g_h(\cdot,\cdot) + r_h(\cdot,\cdot)]$
  $V(\cdot) = \max_{a\in\mathcal{A}} Q(\cdot, a)$.
  $\pi_h(\cdot) = \text{argmax}_{a\in\mathcal{A}} Q_h(\cdot, a)$.
end for
Output: $\{\pi_h\}_{h\in[H]}$.

3.3. Computational Complexity

The runtime complexity of our algorithm in the exploration phase is $\mathcal{O}(HN^4 + H|\mathcal{A}|N^3)$, similar to the runtime complexity in Qiu et al. (2021). At each episode n and each step h, the computation of the kernel ridge regression statistics in each hypercube incurs a cost of $\mathcal{O}(N_c^3 + |\mathcal{A}_c|N_c^2)$, where $|\mathcal{A}_c|$ is the number of actions in the hypercube and N_c is the number of previous observations in the hypercube. Summing up over all hypercubes, we bound the computational complexity with $\mathcal{O}(n^3+|\mathcal{A}|n^2)$, where $|\mathcal{A}|$ is the total number of actions. This bound is derived using the simple arithmetic that the cube of the sum of natural numbers is larger than the sum of their cubes. Summing up over steps and episodes, we arrive at the overall runtime complexity of $\mathcal{O}(HN^4 + H|\mathcal{A}|N^3)$. We expect an improved runtime for our algorithm in practice due to the inequalities used in this calculation.

The computational complexity of identifying the partition with the tightest confidence interval in the planning phase will not exceed that of performing kernel ridge regression on all partitions —a computation similar to that employed in the exploration phase. Therefore, the overall computational complexity remains unaffected by this step.

Sparse approximation methods such as the Nyström method significantly reduce the computational complexity, while maintaining the kernel-based confidence intervals and, consequently, the eventual rates (see, e.g., Vakili et al., 2022, and references therein). These results are generally applicable to kernel ridge regression and not specific to our problem.

4. Sample Complexity Analysis

In this section, we present the main result of the paper. In Theorem 4.5, we establish an $\tilde{\mathcal{O}}\left((\frac{H^3}{\epsilon})^{2+\frac{2d}{\alpha}}\right)$ sample complexity for the kernel-based reward-free RL problem for a general class of kernels with polynomial eigendecay that includes Matérn family and Neural Tangent kernels. The parameter α captures some smoothness properties of the kernel that is specified in the next defintion.

Definition 4.1 (Polynomial Eigendecay)**.** Consider the Mercer eigenvalues $\{\lambda_m\}_{m=1}^\infty$ of $k : \mathcal{Z}\times\mathcal{Z} \to \mathbb{R}$, given in Equation (4), in a decreasing order, as well as the corresponding eigenfeatures $\{\phi_m\}_{m=1}^\infty$. Assume $\mathcal{Z}$ is a d-dimensional hypercube with side length $\rho_\mathcal{Z}$. For some $C_p, \alpha > 0, p > 1$, the kernel k is said to have a polynomial eigendecay, if for all $m \in \mathbb{N}$, $\lambda_m \le C_p m^{-p}\rho_\mathcal{Z}^\alpha$. In addition, for some $\eta \ge 0$, $m^{-p\eta}\phi_m(z)$ is uniformly bounded over all m and z. We

use the notation $\tilde{p} = p(1-2\eta)$.

The polynomial eigendecay profile encompasses a large class of common kernels, e.g., the Matérn family of kernels. For a Matérn kernel with smoothness parameter ν, $p = \frac{2\nu+d}{d}$ and $\alpha = 2\nu$ (e.g., see, Yang et al., 2020a). Another example is the NT kernel (Arora et al., 2019). It has been shown that the RKHS of the NT kernel, when the activations are $s-1$ times differentiable, is equivalent to the RKHS of a Matérn kernel with smoothness $\nu = s - \frac{1}{2}$ (Vakili et al., 2021b). For instance, the RKHS of an NT kernel with ReLU activations is equivalent to the RKHS of a Matérn kernel with $\nu = \frac{1}{2}$. In this case, $p = 1 + \frac{1}{d}$ and $\alpha = 1$. The hypercube domain assumption is a technical formality that can be relaxed to other regular compact subsets of $\mathbb{R}^d$. The uniform boundedness of $m^{-p\eta}\phi_m(z)$ for some $\eta > 0$, also holds for a broad class of kernels, including the Matérn family, as discussed in (Yang et al., 2020a). Several works including (Vakili et al., 2021b; Kassraie & Krause, 2022), have employed an averaging technique over subsets of eigenfeatures, demonstrating that the effective value of η can be considered as 0 in the case of Matérn and NT kernels.

4.1. Confidence Intervals for State-Action Value Functions

Confidence intervals are an important building block in the design and analysis of RL algorithms. For a fixed function f in the RKHS of a known kernel, $1-\delta$ confidence intervals of the form $|f(z) - \mu^{n,f}(z)| \leq \beta(\delta)\sigma^n(z)$ are established in several works (Srinivas et al., 2010; Chowdhury & Gopalan, 2017; Abbasi-Yadkori, 2013; Vakili et al., 2021a) under various assumptions. In the RL setting, however, these confidence intervals cannot be directly applied. This is due to the randomness of the target function itself. Specifically, in our case, the target function is $[P_h V^n_{h+1}]$, which is not a fixed function due to the temporal dependence within an episode. An argument based on the covering number of the state-action value function class was used in Yang et al. (2020a) to establish uniform confidence intervals over all $z \in \mathcal{Z}$ and all f in a specific function class. Vakili & Olkhovskaya (2023) proved a variant that offers flexibility with respect to setting the parameters of the confidence interval. Their approach leads to a more refined confidence interval, which, with a proper choice of parameters, contributes to the improved results in the RL setting.

We first give a formal definition of the two complexity terms: maximum information gain and the covering number of the state-action value function class, which appear in our confidence intervals.

Definition 4.2 (Maximum Information Gain)**.** In the kernel ridge regression setting described in Section 2.3, the following quantity is referred to as maximum information gain: $\Gamma_{k,\tau}(n) = \max_{Z^n \subset \mathcal{Z}} \frac{1}{2}\log\det(I + \frac{1}{\tau^2}K_{Z^n})$.

Upper bounds on maximum information gain based on the spectrum of the kernel are established in (Janz et al., 2020; Srinivas et al., 2010; Vakili et al., 2021c).

State-action value function class: Let us use $\mathcal{Q}_{k,h}(R,B)$ to denote the class of state-action value functions. In particular for a set of observations Z, let $\sigma_h(z)$ be the uncertainty estimate obtained from kernel ridge regression as given in (6). We define

$$\mathcal{Q}_{k,h}(R,B) = \big\{Q : Q(z) = \Pi_{[0,H]}\left\{Q_0(z) + \beta\sigma_h(z)\right\}, \\ \|Q_0\|_{\mathcal{H}_k} \leq R, \beta \leq B, |Z| \leq N\big\}. \tag{17}$$

Definition 4.3 (Covering Set and Number)**.** Consider a function class $\mathcal{F}$. For $\epsilon > 0$, we define the minimum ϵ-covering set $\mathcal{C}(\epsilon)$ as the smallest subset of $\mathcal{F}$ that covers it up to an ϵ error in l_∞ norm. That is to say, for all $f \in \mathcal{F}$, there exists a $g \in \mathcal{C}(\epsilon)$, such that $\|f-g\|_{l_\infty} \leq \epsilon$. We refer to the size of $\mathcal{C}(\epsilon)$ as the ϵ-covering number.

We use the notation $\mathcal{N}_{k,h}(\epsilon; R, B)$ to denote the ϵ-covering number of $\mathcal{Q}_{k,h}(R,B)$, appeaing in the confidence interval.

Lemma 4.4 (Confidence Interval; Theorem 1 of (Vakili & Olkhovskaya, 2023))**.** *Let f^n_h and σ^n_h denote the kernel ridge predictor and uncertainty estimate of $[P_h V^n_{h+1}]$, using n observations $\{V^n_{h+1}(s^i_{h+1})\}_{i=1}^n$ at $Z^n_h = \{z^i_h\}_{i=1}^n \subset \mathcal{Z}$, where s^i_{h+1} is the next state drawn from $P_h(\cdot|z^i_h)$. Let $R_N = H + \frac{H}{2\lambda}\sqrt{2(\Gamma_{k,\tau}(N) + 1 + \log(\frac{2}{\delta}))}$. For $\epsilon, \delta \in (0,1)$, with probability, at least $1-\delta$, we have, $\forall (z,h) \in \mathcal{Z}\times[H]$ and $n \in [N]$,*

$$\left|[P_h V^n_{h+1}](z) - f^n_h(z)\right| \leq \beta^n_h(\delta,\epsilon)\sigma^n_h(z) + \epsilon,$$

where $\beta^n_h(\delta,\epsilon)$ is set to any value satisfying

$$\beta^n_h(\delta,\epsilon) \geq H + \frac{3\sqrt{n}\epsilon}{\tau} + \frac{H}{\sqrt{2}}\times \\ \sqrt{\Gamma_{k,\tau}(n) + \log\mathcal{N}_{k,h}(\epsilon; R_N, \beta^n_h(\delta,\epsilon)) + 1 + \log(\frac{2NH}{\delta})}.$$

4.2. Sample Complexity

With the auxiliary results laid out in the previous section, we now give a formal presentation of sample complexity.

Theorem 4.5. *Consider the reward-free kernel-based RL problem presented in Section 2. Under Assumption 2.2; run Algorithm 1 with N exploration episodes. Let π be the policy obtained in the planning phase according to Algorithm 2. There exist $N = \tilde{\mathcal{O}}\left((\frac{H^3}{\epsilon})^{2+\frac{2d}{\alpha}}\right)$ and $\beta(\delta) = \mathcal{O}(H\sqrt{\log(\frac{HN}{\delta})})$, which guarantee, with probability at least $1-\delta$, we have, $V^\star(s) - V^\pi(s) \leq \epsilon$, for all $s \in \mathcal{S}$.*

Proof. Here, we present a summary of the steps in the proof. Details are provided in Appendix B. In the proof, we show that $V_1^\star(s;r) - V_1^\pi(s;r) \leq \frac{H}{N}\sum_{n=1}^N V_1^\star(s,\tilde{r}^n)$, that creates a connection between suboptimality in the planning (the left hand side) and total value in the exploration with hypothetical rewards (the right hand side). We then use confidence intervals for kernel-based regression over partitions to bound the right hand side, and eventually obtain

$$V_1^\star(s;r) - V_1^\pi(s_1;r) = \mathcal{O}\left(N^{\frac{-\alpha}{2(\alpha+d)}}\log(N)\sqrt{\log(\frac{H}{\delta})}\right).$$

From here we can see that with a choice of $N = \tilde{\mathcal{O}}\left((\frac{H^3}{\epsilon})^{2+\frac{2d}{\alpha}}\right)$ it can be guaranteed that $V_1^\star(s;r) - V_1^\pi(s_1;r) \leq \epsilon$. □

Evaluation of the result. When we instantiate the sample complexity for the Matérn kernel, we obtain a sample complexity of $\tilde{\mathcal{O}}\left((\frac{H^3}{\epsilon})^{2+\frac{d}{\nu}}\right)$. This is order optimal in ϵ given the $\Omega\left((\frac{1}{\epsilon})^{2+\frac{d}{\nu}}\right)$ sample complexity lower bound for kernel-based bandits with Matérn kernels (see, Scarlett et al., 2017, *Table I*), up to logarithmic factors in ϵ. We note that kernel-based bandit corresponds to a degenerate case with $H = 1, |\mathcal{S}| = 1$. Thus, this cannot be further improved. In addition, our sample complexity in its dependency on ϵ matches that of the simpler discounted kernel-based RL problem assuming the existence of a generative oracle that can provide arbitrary transition samples (see, Yeh et al., 2023, *Table* 1).

Regarding the dependency of sample complexity on episode length H, however, it is still unresolved whether improvements are possible. For comparison, in the oracle setting (Yeh et al., 2023), the sample complexity is proportional to $(\frac{1}{1-\gamma})^{1+3(2+\frac{d}{\nu})}$, where $\gamma \in (0,1)$ is the discount factor in the discounted MDP setting. Informally interpreting $\frac{1}{1-\gamma}$ as the *effective* episode length, our results show a similar dependency on H. Notably, our sample complexity reflects and improvement by a factor of H.

5. Conclusion

We considered the reward-free RL framework with kernel-based modeling. We developed algorithms for both exploration and planning phases. Our results shows an order optimal sample complexity for a general class of common kernels with polynomially decaying eigenvalues, that includes Matérn and Neural Tangent kernels. We significantly improve the state of the art as the existing work does not apply to this class of kernels (with an unbounded sample complexity). In addition, the scaling of the sample complexity in ϵ matches that of the lower bound in kernel-based bandits with Matérn kernel, showing its optimality.

Impact Statement

This paper presents research aimed at advancing the field of Machine Learning. While there are numerous potential societal implications associated with RL, we do not believe any require specific emphasis given the theoretical nature of the work.

References

Abbasi-Yadkori, Y. Online learning for linearly parametrized control problems. *PhD Thesis, University of Alberta*, 2013.

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, volume 24, 2011.

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

Agrawal, S. and Goyal, N. Thompson sampling for contextual bandits with linear payoffs. In *International conference on machine learning*, pp. 127–135. PMLR, 2013.

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R. R., and Wang, R. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32, 2019.

Auer, P., Jaksch, T., and Ortner, R. Near-optimal regret bounds for reinforcement learning. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008.

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

Bartlett, P. L. and Tewari, A. REGAL: A regularization based algorithm for reinforcement learning in weakly communicating mdps. *CoRR*, abs/1205.2661, 2012. URL http://arxiv.org/abs/1205.2661.

Borovitskiy, V., Terenin, A., Mostowsky, P., et al. Matérn Gaussian processes on Riemannian manifolds. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12426–12437, 2020.

Chen, X., Hu, J., Yang, L. F., and Wang, L. Near-optimal reward-free exploration for linear mixture mdps with plug-in solver. *arXiv preprint arXiv:2110.03244*, 2021.

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

Chowdhury, S. R. and Gopalan, A. Online learning in kernelized Markov decision processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3197–3205. PMLR, 2019.

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

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

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. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53, 2022.

Janz, D., Burt, D., and González, J. Bandit optimisation of functions in the Matérn kernel RKHS. In *International Conference on Artificial Intelligence and Statistics*, pp. 2486–2495. PMLR, 2020.

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

Jin, C., Krishnamurthy, A., Simchowitz, M., and Yu, T. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pp. 4870–4879. PMLR, 2020a.

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

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

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

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

Kassraie, P. and Krause, A. Neural contextual bandits without regret. In *International Conference on Artificial Intelligence and Statistics*, pp. 240–278. PMLR, 2022.

Kaufmann, E., Ménard, P., Domingues, O. D., Jonsson, A., Leurent, E., and Valko, M. Adaptive reward-free exploration. In *Algorithmic Learning Theory*, pp. 865–891. PMLR, 2021.

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

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

Ménard, P., Domingues, O. D., Jonsson, A., Kaufmann, E., Leurent, E., and Valko, M. Fast active learning for pure exploration in reinforcement learning. In *International Conference on Machine Learning*, pp. 7599–7608. PMLR, 2021.

Mercer, J. 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, 1909. ISSN 02643952. URL http://www.jstor.org/stable/91043.

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

Neu, G. and Pike-Burke, C. A unifying view of optimism in episodic reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1392–1403, 2020.

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

Qiu, S., Ye, J., Wang, Z., and Yang, Z. On reward-free rl with kernel and neural function approximations: Single-agent mdp and markov game. In *International Conference on Machine Learning*, pp. 8737–8747. PMLR, 2021.

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

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

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

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

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

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

Vakili, S. and Olkhovskaya, J. Kernelized reinforcement learning with order optimal regret bounds. *arXiv preprint arXiv:2306.07745*, 2023.

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

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

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

Vakili, S., Scarlett, J., Shiu, D.-s., and Bernacchia, A. Improved convergence rates for sparse approximation methods in kernel-based learning. In *International Conference on Machine Learning*, pp. 21960–21983. PMLR, 2022.

Valko, M., Korda, N., Munos, R., Flaounas, I., and Cristianini, N. Finite-time analysis of kernelised contextual bandits. In *Uncertainty in Artificial Intelligence*, 2013.

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

Wagenmaker, A. J., Chen, Y., Simchowitz, M., Du, S., and Jamieson, K. Reward-free RL is no harder than reward-aware RL in linear markov decision processes. In *International Conference on Machine Learning*, pp. 22430–22456. PMLR, 2022.

Wang, R., Du, S. S., Yang, L., and Salakhutdinov, R. R. On reward-free reinforcement learning with linear function approximation. *Advances in neural information processing systems*, 33:17816–17826, 2020.

Williams, C. K. and Rasmussen, C. E. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.

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

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

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

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

Yao, H., Szepesvári, C., Pires, B. A., and Zhang, X. Pseudo-MDPs and factored linear action models. In *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pp. 1–9. IEEE, 2014.

Yeh, S.-Y., Chang, F.-C., Yueh, C.-W., Wu, P.-Y., Bernacchia, A., and Vakili, S. Sample complexity of kernel-based q-learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 453–469. PMLR, 2023.

Zanette, A., Brandfonbrener, D., Brunskill, E., Pirotta, M., and Lazaric, A. Frequentist regret bounds for randomized least-squares value iteration. In *International Conference on Artificial Intelligence and Statistics*, pp. 1954–1964. PMLR, 2020a.

Zanette, A., Lazaric, A., Kochenderfer, M., and Brunskill, E. Learning near optimal policies with low inherent Bellman error. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10978–10989. PMLR, 13–18 Jul 2020b.

Zanette, A., Lazaric, A., Kochenderfer, M. J., and Brunskill, E. Provably efficient reward-agnostic navigation with linear value iteration. *Advances in Neural Information Processing Systems*, 33:11756–11766, 2020c.

Zhang, J., Zhang, W., and Gu, Q. Optimal horizon-free reward-free exploration for linear mixture mdps. In *International Conference on Machine Learning*, pp. 41902–41930. PMLR, 2023.

Zhang, W., Zhou, D., and Gu, Q. Reward-free model-based reinforcement learning with linear function approximation. *Advances in Neural Information Processing Systems*, 34:1582–1593, 2021.

Zhang, Z., Du, S. S., and Ji, X. Nearly minimax optimal reward-free reinforcement learning. *arXiv preprint arXiv:2010.05901*, 2020.

A. RKHS and Mercer Theorem

Mercer theorem (Mercer, 1909) provides a representation of the kernel in terms of an infinite dimensional feature map (e.g., see, Christmann & 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

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

Theorem A.1. *(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-eigenfeature pairs $\{(\lambda_m, \phi_m)\}_{m=1}^{\infty}$ such that $\lambda_m > 0$, and $T_k \phi_m = \lambda_m \phi_m$, for $m \geq 1$. Moreover, the kernel function can be represented as

$$k\left(z, z'\right) = \sum_{m=1}^{\infty} \lambda_m \phi_m(z) \phi_m\left(z'\right),$$

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

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

Theorem A.2. *(Mercer Representation Theorem) Let $\{(\lambda_m, \phi_m)\}_{i=1}^{\infty}$ be the Mercer eigenvalue-eigenfeature pairs. Then, the RKHS of k is given by*

$$\mathcal{H}_k = \left\{ f(\cdot) = \sum_{m=1}^{\infty} w_m \lambda_m^{\frac{1}{2}} \phi_m(\cdot) : w_m \in \mathbb{R}, \|f\|^2_{\mathcal{H}_k} := \sum_{m=1}^{\infty} w_m^2 < \infty \right\}.$$

Mercer representation theorem indicates that the scaled eigenfeatures $\{\sqrt{\lambda_m}\phi_m\}_{m=1}^{\infty}$ form an orthonormal basis for $\mathcal{H}_k$.

B. Detailed Analysis of Sample Complexity

Recall the reward-free kernel-based RL problem formulated in Section 2. In the exploration phase, trajectories are collected for N episodes based on Algorithm 1. In the exploitation phase, after the reward r is revealed, a policy π is obtained according to Algorithm 2. We establish an upper bound on $V_1^\star(s;r) - V_1^\pi(s;r)$, where $V_h^\star(s;r)$ is the optimal value function with r and $V_h^\pi(s;r)$ is the value function for the policy π. It is then straightforward to obtain the appropriate sample complexity N which guarantees $V_1^\star(s;r) - V_1^\pi(s;r) \leq \epsilon$.

Proof Structure. We structure our proof as follows. We first recall some notations, prove an upper bound on the total number of partition components created by the algorithm in the exploration phase, and introduce two high probability events based on confidence intervals. We then present two main steps of the proof in Sections B.1 and B.2, respectively. In the first step, we bound the suboptimality of the policy in the planning phase using the total value function in the exploration phase. In the second step, we bound the total value function in the exploration phase. The proof of lemmas is given in Appendix C.

Recall the notations $V_h^\pi(s;r)$ and $Q_h^\pi(s;r)$ for the value function and state-action value function of policy π with reward r. In contrast, V_h^n and Q_h^n represent the proxies for value function and state-action value function used in the episode n of exploration phase (Algorithm 1), and V_h and Q_h are those used in the planning phase (Algorithm 2).

Next, we bound the total number of partitions used in the exploration phase by the algorithm. This result is used in several places in the proof. For step h, let $\mathcal{U}_h^N = \bigcup_{n=1}^N \mathcal{S}_h^N$ be the union of all cover elements used by the algorithm over all exploration episodes. The size of $\mathcal{U}_h^N$ is bounded in the following lemma.

Lemma B.1 (Lemma 2 in (Janz et al., 2020)). *The size of $\mathcal{U}_h^N$ satisfies*

$$|\mathcal{U}_h^N| \leq C_1 N^{\frac{d}{d+\alpha}}, \tag{18}$$

for some constant $C_1 > 0$.

The size of $\mathcal{U}_h^N$ depends on the dimension of the domain and the parameter α used in the splitting rule of domain partitioning.

Let us define event $\mathcal{E}$ as the event that all the kernel-based confidence intervals in the exploration phase hold true; i.e., $\forall z \in \mathcal{Z}, \forall (h,n) \in [H] \times [N]$,

$$\left|[P_h V_{h+1}^n](z) - f_h^n(z)\right| \leq \beta(\delta)\sigma_h^n(z), \tag{19}$$

where f_h^n and σ_h^n are the kernel ridge predictor and uncertainty estimates of $[P_h V_{h+1}^n]$ defined in Equations (11) and (12), respectively.

Similarly, let us define event $\tilde{\mathcal{E}}$ as the event that all the kernel-based confidence intervals in the planning phase hold true; i.e., $\forall z \in \mathcal{Z}, \forall (h,n) \in [H] \times [N]$,

$$|[P_h V_{h+1}](z) - g_h(z)| \leq w_h, \tag{20}$$

where g_h and w_h are defined in Equations (16) and (15), respectively.

Lemma B.2. *With a choice of $\beta(\delta) = \mathcal{O}(H\sqrt{\log(\frac{HN}{\delta})})$ with a sufficiently large implied constant, the events $\mathcal{E}$ and $\tilde{\mathcal{E}}$ each hold with probability at least $1 - \delta/3$: $\Pr(\mathcal{E}) \geq 1 - \delta/3$ and $\Pr(\tilde{\mathcal{E}}) \geq 1 - \delta/3$.*

B.1. Suboptimality of the Planning Phase

The goal of this section of the proof is to bound $V_1^\star(s;r) - V_1^\pi(s;r)$ using $\sum_{n=1}^N V_1^\star(s;\tilde{r}^n)$. This creates the connection between the planning and exploration phases. To bound $V_1^\star(s;r) - V_1^\pi(s;r)$, we bound the following two terms in the following two lemmas: $V_1^\star(s;r) - V_1(s)$ and $V_1(s) - V_1^\pi(s;r)$.

Lemma B.3. *Conditioned on $\tilde{\mathcal{E}}$, we have*

$$V_h^\star(s_1;r) - V_h(s_1) \leq 0.$$

Lemma B.4. *Conditioned on $\tilde{\mathcal{E}}$, we have*

$$V_1(s_1) - V_1^\pi(s_1;r) \leq \sum_{h=1}^H w_h(s_h, \pi(s_h)).$$

Combining Lemmas B.3 and B.4, we obtain the following

$$V_1^\star(s_1;r) - V_1^\pi(s_1;r) \leq \sum_{h=1}^H w_h(s_h, \pi(s_h)). \tag{21}$$

By definition of w_h, we have $w_h(z) \leq \beta(\delta)\sigma_h^n(z)$ for all $n \leq N$. Recall definition of $\tilde{r}^n = \beta(\delta)\sigma^n/H$. We have

$$\sum_{h=1}^H w_h(s_h, \pi(s_h)) \leq H V_1^\star(s_1;\tilde{r}^n) \tag{22}$$

Summing up over n and dividing by N

$$\sum_{h=1}^H w_h(s_h, \pi(s_h)) \leq \frac{H}{N}\sum_{n=1}^N V_1^\star(s;\tilde{r}^n). \tag{23}$$

From (21) and (23), we can see that

$$V_1^\star(s_1;r) - V_1^\pi(s_1;r) \le \frac{H}{N}\sum_{n=1}^{N} V_1^\star(s;\tilde{r}^n). \tag{24}$$

This connect the suboptimality for the planning phase to the total value of the exploration phase.

B.2. Total Value Function in the Exploration Phase

In this section, our goal is to bound $\sum_{n=1}^N V_1^\star(s;\tilde{r}^n)$. We first bound $V_h^\star(s;\tilde{r}^n) - V_h^n(s)$ in Lemma B.5, and then bound the $\sum_{n=1}^N V_1^n(s)$ in Lemma B.6.

Lemma B.5. *Conditioned on $\mathcal{E}$, we have*

$$V_h^\star(s;\tilde{r}^n) - V_h^n(s) \le 0.$$

Lemma B.6. *Define $\zeta_h^n = [P_h V_{h+1}^n](s_h^n, a_h^n) - V_{h+1}(s_{h+1}^n)$. Conditioned on $\mathcal{E}$, we have*

$$V_1^n(s) \le \sum_{h=1}^{H} \zeta_h^n + (2+\frac{1}{H})\sum_{h=1}^{H}\beta(\delta)\sigma_h^n(s_h^n,a_h^n).$$

Summing both sides over n, we obtain

$$\sum_{n=1}^{N} V_1^n(s) \le \sum_{n=1}^{N}\sum_{h=1}^{H} \zeta_h^n + (2+\frac{1}{H})\sum_{n=1}^{N}\sum_{h=1}^{H}\beta(\delta)\sigma_h^n(s_h^n,a_h^n).$$

Using Lemma B.5, we get

$$\sum_{n=1}^{N} V_1^\star(s;\tilde{r}^n) \le \underbrace{\sum_{n=1}^{N}\sum_{h=1}^{H}\zeta_h^n}_{\text{Term 1}} + \underbrace{(2+\frac{1}{H})\sum_{n=1}^{N}\sum_{h=1}^{H}\beta(\delta)\sigma_h^n(s_h^n,a_h^n)}_{\text{Term 2}}. \tag{25}$$

We next bound the two terms on the right hand side.

Term 1. By Azuma-Hoeffding inequality with probability at least $1-\delta/3$,

$$\sum_{n=1}^{N}\sum_{h=1}^{H}\zeta_h^n \le \mathcal{O}\left(\sqrt{H^3 N \log(\frac{1}{\delta})}\right). \tag{26}$$

Term 2. We bound the total uncertainty in the kernel ridge regression measured by $\sum_{n=1}^N (\sigma_h^n(z_h^n))^2$

$$\begin{aligned}
\sum_{n=1}^{N}(\sigma_h^n(z_h^n))^2 &= \sum_{\mathcal{Z}'\in\mathcal{U}_h^N}\sum_{z_h^n\in\mathcal{Z}'}(\sigma_h^n(z_h^n))^2 \\
&\le \sum_{\mathcal{Z}'\in\mathcal{U}_h^N}\frac{2}{\log(1+1/\lambda^2)}\Gamma_{k,\lambda}(N_{h,\mathcal{Z}'}^N) \\
&= \mathcal{O}\left(\sum_{\mathcal{Z}'\in\mathcal{U}_h^N}\log(N)\right) \\
&= \mathcal{O}\left(|\mathcal{U}_h^N|\log(N)\right) \\
&= \mathcal{O}\left(N^{\frac{d}{d+\alpha}}\log(N)\right).
\end{aligned}$$

The first inequality is commonly used in kernelized bandits. For example see Srinivas et al. (2010), Lemma 5.4. The third and fifth lines follow from Equation (33) in Appendix C, and Lemma B.1, respectively.

Therefore, using Cauchy-Schwarz inequality,

$$\sum_{n=1}^{N} \sigma_h^n(z_h^n) \leq \sqrt{N \sum_{n=1}^{N} \left(\sigma_h^n(z_h^n)\right)^2} \leq \mathcal{O}\left(N^{\frac{d+\alpha/2}{d+\alpha}} \sqrt{\log(N)}\right). \tag{27}$$

Replacing the value for $\beta(\delta)$ and summing over h, we obtain

$$\text{Term 2} = \mathcal{O}\left(H^2 N^{\frac{d+\alpha/2}{d+\alpha}} \sqrt{\log(\frac{NH}{\delta})\log(N)}\right). \tag{28}$$

Combining the bound on two terms, we get

$$\sum_{n=1}^{N} V_1^\star(s;\tilde{r}^n) = \mathcal{O}\left(H^2 N^{\frac{d+\alpha/2}{d+\alpha}} \sqrt{\log(\frac{NH}{\delta})\log(N)}\right). \tag{29}$$

B.3. Sample Complexity

From Equations (24) and (29), proven in the previous sections, we have, with probability at least $1-\delta$

$$\begin{aligned} V_1^\star(s_1;r) - V_1^\pi(s_1;r) &= \mathcal{O}\left(\frac{H^3}{N} N^{\frac{d+\alpha/2}{d+\alpha}} \sqrt{\log(\frac{NH}{\delta})\log(N)}\right) \\ &= \mathcal{O}\left(H^3 N^{\frac{-\alpha}{2(\alpha+d)}} \log(N) \sqrt{\log(\frac{H}{\delta})}\right). \end{aligned} \tag{30}$$

Then, the choice of

$$N = \Theta\left((\frac{H^3\sqrt{\log(\frac{H}{\delta})}}{\epsilon})^{2+\frac{2d}{\alpha}} \mathit{polylog}(\frac{H^3\sqrt{\log(\frac{H}{\delta})}}{\epsilon})\right) \tag{31}$$

with a sufficiently large constant, ensures that $V_1^\star(s_1;r) - V_1^\pi(s_1;r) \leq \epsilon$; i.e, the policy π obtained in the planning phase is ϵ-optimal.

C. Proof of Lemmas

In this section we provide the proof of lemmas.

C.1. Proof of Lemma B.2

The lemma is a result of confidence intervals given in Lemma 4.4. We only need to prove that $\beta(\delta)$ given in Theorem 4.5 satisfies the condition on the confidence interval width multiplier given in Lemma 4.4.

Consider a cover element $\mathcal{Z}' \in \mathcal{U}_h^N$. Using Lemma 4.4, we have, with probability at least $1-\delta$, for all $h \in [H], n \in [N], z \in \mathcal{Z}'$, for some $\epsilon_h^n \in (0,1)$,

$$\left|[P_h V_{h+1}^n](z) - f_h^n(z)\right| \leq \beta_h^n(\delta,\epsilon_h^n)\sigma_h^n(z) + \epsilon_h^n, \tag{32}$$

where $\beta_h^n(\delta,\epsilon_h^n)$ is the smallest value satisfying

$$\beta_h^n(\delta,\epsilon_h^n) \geq H + 1 + \frac{H}{\sqrt{2}}\sqrt{\Gamma_{k,\tau}(N) + \log \mathcal{N}_{k,h}(\epsilon_h^n; R_N, \beta_h^n(\delta,\epsilon_h^n)) + 1 + \log(\frac{NH}{\delta})} + \frac{3\sqrt{N}\epsilon_h^n}{\tau},$$

with $N = N^n_{h,\mathcal{Z}'}$ and $\epsilon^n_h = \frac{H\sqrt{\log(\frac{HN}{\delta})}}{\sqrt{N^n_{h,\mathcal{Z}'}}}$.

We use the following lemma to bound the maximum information gain term.

Lemma C.1 (Lemma 2 in (Vakili & Olkhovskaya, 2023)). *Consider a positive definite kernel $k : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$, with polynomial eigendecay on a hypercube with side length $\rho_{\mathcal{Z}}$. The maximum information gain given in Definition 4.2 satisfies*

$$\Gamma_{k,\tau}(T) = \mathcal{O}\left(N^{\frac{1}{p}}(\log(N))^{1-\frac{1}{p}}\rho_{\mathcal{Z}}^{\frac{\alpha}{p}}\right).$$

Therefore,

$$\begin{aligned}
\Gamma_{k,\tau}(N^n_{h,\mathcal{Z}'}) &= \mathcal{O}\left((N^n_{h,\mathcal{Z}'})^{\frac{1}{p}}(\log(N^n_{h,\mathcal{Z}'}))^{1-\frac{1}{p}}\rho_{\mathcal{Z}'}^{\frac{\alpha}{p}}\right) \\
&= \mathcal{O}\left((\rho_{\mathcal{Z}'})^{\frac{-\alpha}{p}}(\log(N^n_{h,\mathcal{Z}'}))^{1-\frac{1}{p}}\rho_{\mathcal{Z}'}^{\frac{\alpha}{p}}\right) \\
&= \mathcal{O}\left((\log(N^n_{h,\mathcal{Z}'}))^{1-\frac{1}{p}}\right) \\
&= \mathcal{O}\left(\log(N)\right), \qquad (33)
\end{aligned}$$

where the first line is based on Lemma C.1, and the second line is by the design of partitioning in the exploration algorithm. Recall that each hypercube is partitioned when $\rho^{-\alpha}_{\mathcal{Z}'} < N^n_{h,\mathcal{Z}'} + 1$, ensuring that $N^n_{h,\mathcal{Z}'}$ remains at most $\rho^{-\alpha}_{\mathcal{Z}'}$.

We use the following lemma to bound the covering number of the space of functions.

Lemma C.2 (Lemma 3 in (Vakili & Olkhovskaya, 2023)). *Recall the class of state-action value functions $\mathcal{Q}_{k,h}(R,B)$, where $k : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ satisfies the polynomial eigendecay on a hypercube with side length $\rho_{\mathcal{Z}}$. We have*

$$\log \mathcal{N}_{k,h}(\epsilon; R, B) = \mathcal{O}\left(\frac{R^2\rho_{\mathcal{Z}}^{\alpha}}{\epsilon^2}^{\frac{1}{\bar{p}-1}} 1 + \log\frac{R}{\epsilon} + \frac{B^2\rho_{\mathcal{Z}}^{\alpha}}{\epsilon^2}^{\frac{2}{\bar{p}-1}} 1 + \log\frac{B}{\epsilon}\right).$$

For the covering number, with the choice of $\epsilon^n_h = \frac{H\sqrt{\log(\frac{HN}{\delta})}}{\sqrt{N^n_{h,\mathcal{Z}'}}}$, we have

$$\begin{aligned}
&\log \mathcal{N}_{k,h}(\epsilon^n_h; R_N, \beta^n_h(\delta, \epsilon^n_h)) \\
&= \mathcal{O}\left(\left(\frac{R_N^2\rho_{\mathcal{Z}'}^{\alpha}}{(\epsilon^n_h)^2}\right)^{\frac{1}{\bar{p}-1}}(1+\log(\frac{R_N}{\epsilon^n_h})) + \left(\frac{(\beta^n_h(\delta,\epsilon^n_h))^2\rho_{\mathcal{Z}'}^{\alpha}}{(\epsilon^n_h)^2}\right)^{\frac{2}{\bar{p}-1}}(1+\log(\frac{\beta^n_h(\delta,\epsilon^n_h)}{\epsilon^n_h}))\right) \\
&= \mathcal{O}\left(\left(\frac{R_N^2}{H^2\log(\frac{HN}{\delta})}\right)^{\frac{1}{\bar{p}-1}}(1+\log(\frac{R_N}{\epsilon^n_h})) + \left(\frac{(\beta^n_h(\delta,\epsilon^n_h))^2}{H^2\log(\frac{HN}{\delta})}\right)^{\frac{2}{\bar{p}-1}}(1+\log(\frac{\beta^n_h(\delta,\epsilon^n_h)}{\epsilon^n_h}))\right).
\end{aligned}$$

We thus see that the choice of $\beta^n_h(\delta,\epsilon^n_h) = \Theta(H\sqrt{\log(\frac{HN}{\delta})})$ satisfies the requirement for confidence interval width on $\mathcal{Z}'$. We now use probability union bound over all $\mathcal{Z}' \in \mathcal{U}^n_h$ to obtain

$$\beta(\delta) = \Theta(H\sqrt{\log(\frac{HN|H\mathcal{U}^n_h|}{\delta})}) = \Theta(H\sqrt{\log(\frac{HN}{\delta})}). \qquad (34)$$

For this value of $\beta(\delta)$, we have with probability at least $1-\delta$, for all $h \in [H], n \in [N], z \in \mathcal{Z}$,

$$\left|[P_hV^n_{h+1}](z) - f^n_h(z)\right| \le \beta(\delta)\sigma^n_h(z) + \epsilon^n_h, \qquad (35)$$

where in the above expression ϵ^n_h is the parameter of the covering number corresponding to $\mathcal{Z}'$ when $z \in \mathcal{Z}'$. Thus $\Pr(\mathcal{E}) \ge 1-\delta$. Finally since $\epsilon^n_h = \frac{H\sqrt{\log(\frac{HN}{\delta})}}{\sqrt{N^n_{h,\mathcal{Z}'}}}$ is always smaller than the first term, we have

$$\left|[P_hV^n_{h+1}](z) - f^n_h(z)\right| \le \beta(\delta)\sigma^n_h(z). \qquad (36)$$

where β is multiplied with a factor of 2 that does not affect its expression in Equation (34).

The proof of $\Pr(\tilde{\mathcal{E}}) \ge 1-\delta$ is similar.

C.2. Proof of Lemma B.3

This can be proven by induction. For $h = H + 1$, we have $V^\star_{H+1}(s; r) = V_{H+1}(s) = 0$, for any $s \in \mathcal{S}$. Also, conditioned on $\tilde{\mathcal{E}}$, we have

$$\begin{aligned} Q^\star_h(z; r) - Q_h(z) &= r_h(z) + [P_h V^\star_{h+1}](z; r) - \min\{r_h(z) + g_h(z) + w_h(z), H\} \\ &= \max\left\{[P_h V^\star_{h+1}](z; r) - g_h(z) - w_h(z), 0\right\} \\ &= \max\left\{[P_h V^\star_{h+1}](z; r) - [P_h V_{h+1}](z; r) + [P_h V_{h+1}](z; r) - g_h(z) - w_h(z), 0\right\} \\ &\le \max\left\{[P_h V^\star_{h+1}](z; r) - [P_h V_{h+1}](z; r), 0\right\} \\ &\le 0. \end{aligned}$$

The first inequality comes form event $\tilde{\mathcal{E}}$ and the second inequality comes form induction assumption. Then, we have

$$\begin{aligned} V^\star_h(s; r) - V_h(s) &= \max_{a\in\mathcal{A}} Q^\star_h(z; r) - \max_{a\in\mathcal{A}} Q_h(z) \\ &\le \max_{a\in\mathcal{A}}\{Q^\star_h(z; r) - Q_h(z)\} \\ &\le 0. \end{aligned}$$

That completes the proof.

C.3. Proof of Lemma B.4

We prove this by induction. We have $V_{H+1}(s) = V^\pi_{H+1}(s) = 0$, for any $s \in \mathcal{S}$. Also, conditioned on $\tilde{\mathcal{E}}$, we have

$$\begin{aligned} V_h(s_h) - V^\pi_h(s_h; r) &= r_h(s_h, \pi_h(s_h)) + g_h(s_h, \pi_h(s_h)) + w_h(s_h, \pi_h(s_h)) - Q^\pi_h(s_h, \pi_h(s_h); r) \\ &\le r_h(s_h, \pi_h(s_h)) + [P_h V_{h+1}](s_h, \pi_h(s_h)) + 2w_h(s_h, \pi_h(s_h)) - r_h(s_h, \pi_h(s_h)) - [P_h V^\pi_{h+1}](s_h, \pi_h(s_h); r) \\ &= [P_h V_{h+1}](s_h, \pi_h(s_h)) - [P_h V^\pi_{h+1}](s_h, \pi_h(s_h); r) + 2w_h(s_h, \pi_h(s_h)) \\ &= \mathbb{E}_{s_{h+1}\sim P_h(\cdot|s_h,\pi_h(s_h))}[V_{h+1}(s_{h+1}) - V^\pi_{h+1}(s_{h+1})] + 2w_h(s_h, \pi_h(s_h)) \\ &\le \sum_{h'=h}^{H} w_h(s_h, \pi(s_h)). \end{aligned}$$

The first inequality comes form event $\tilde{\mathcal{E}}$ and the second inequality comes form induction assumption.

C.4. Proof of Lemma B.5

This can be proven by induction. For $h = H + 1$, we have $V^\star_{H+1}(s; \tilde{r}^n) = V^n_{H+1}(s) = 0$, for any $s \in \mathcal{S}$. Also, conditioned on $\mathcal{E}$, we have

$$\begin{aligned} Q^\star_h(z; \tilde{r}^n) - Q^n_h(z) &= \tilde{r}^n_h(z) + [P_h V^\star_{h+1}](z) - \min\{\tilde{r}^n_h(z) + f^n_h(z) + \beta(\delta)\sigma^n_h(z), H\} \\ &= \max\left\{[P_h V^\star_{h+1}](z) - f^n_h(z) - \beta(\delta)\sigma^n_h(z), 0\right\} \\ &\le \max\left\{[P_h V^\star_{h+1}](z) - [P_h V^n_{h+1}](z), 0\right\} \\ &\le 0. \end{aligned}$$

The first inequality holds under event $\mathcal{E}$, and the second inequality comes from induction assumption.

C.5. Proof of Lemma B.6

Recall $\zeta_h^n = [P_h V_{h+1}^n](s_h^n, a_h^n) - V_{h+1}(s_{h+1}^n)$. We have, under event $\mathcal{E}$

$$\begin{aligned} V_h^n(s_h^n) &= Q_h^n(s_h^n, a_h^n) \\ &= f_h^n(s_h^n, a_h^n) + \tilde{r}_h^n(s_h^n, a_h^n) + \beta(\delta)\sigma_h^n(s_h^n, a_h^n) \\ &\leq [P_h V_{h+1}^n](s_h^n, a_h^n) + \tilde{r}_h^n(s_h^n, a_h^n) + 2\beta(\delta)\sigma_h^n(s_h^n, a_h^n) \\ &= [P_h V_{h+1}^n](s_h^n, a_h^n) + (2 + \frac{1}{H})\beta(\delta)\sigma_h^n(s_h^n, a_h^n) \\ &= \zeta_h^n + V_{h+1}^n(s_{h+1}^n) + (2 + \frac{1}{H})\beta(\delta)\sigma_h^n(s_h^n, a_h^n) \end{aligned}$$

The inequality holds under event $\mathcal{E}$. Summing the telescoping series

$$V_h^n(s_h^n) - V_{h+1}^n(s_{h+1}^n) \leq \zeta_h^n + (2 + \frac{1}{H})\beta(\delta)\sigma_h^n(s_h^n, a_h^n)$$

over h, we get

$$V_1^n(s) \leq \sum_{h=1}^{H} \zeta_h^n + (2 + \frac{1}{H}) \sum_{h=1}^{H} \beta(\delta)\sigma_h^n(s_h^n, a_h^n).$$

Taking summation over n

$$\sum_{n=1}^{N} V_1^n(s) \leq \sum_{n=1}^{N}\sum_{h=1}^{H} \zeta_h^n + (2 + \frac{1}{H}) \sum_{n=1}^{N}\sum_{h=1}^{H} \beta(\delta)\sigma_h^n(s_h^n, a_h^n),$$

that completes the proof.