LEARNING TWO-PLAYER MIXTURE MARKOV GAMES: KERNEL FUNCTION APPROXIMATION AND CORRELATED EQUILIBRIUM

Anonymous authors
Paper under double-blind review

ABSTRACT

We consider learning the Nash equilibrium in two-player Markov Games with nonlinear function approximation, where the action-value function is approximated by a function in the Reproducing Kernel Hilbert Space (RKHS) space. The key challenge is how to do exploration in the high-dimensional function space. We propose novel online learning algorithms to find the Nash Equilibrium by minimizing the duality gap. At the core of our algorithms are upper and lower confidence bounds that are derived based on the principle of optimism in the face of uncertainty. We prove that our algorithm is able to attain an $O(\sqrt{T})$ regret with polynomial computational complexity, under very mild assumptions on the reward function and the underlying dynamic of the Markov games. This work provides one of the first results of desirable complexities in the learning of two-player Markov games with nonlinear function approximation in the kernel mixture settings, and its implications for function approximation via deep neural networks.

1 INTRODUCTION

Multi-agent reinforcement learning (MARL) has been the focus of research across a range of research communities (Shapley, 1953; Littman, 1994). The case of two-player Markov games (MG) has been of particular interest. In this case, two players select their actions based on the current state simultaneously and independently. One player (the max-player) aims to maximize the return based on the reward provided by the environment, while the other (the min-player) aims to minimize it. A series of recent results have established polynomial sample complexity/regret dependence on the cardinality of state/action spaces for two-player MG (Wei et al., 2017; Bai & Jin, 2020; Bai et al., 2020; Liu et al., 2020; Jia et al., 2019; Sidford et al., 2020; Cui & Yang, 2020; Lagoudakis & Parr, 2002; Perolat et al., 2015; Pérolat et al., 2016a;b; 2017).

Meanwhile, most of the successful recent empirical applications of MARL deal with *large state/action spaces* that may be continuous or a fine-grained discretization of a continous space. Examples include GO (Silver et al., 2016), autonomous driving (Shalev-Shwartz et al., 2016), TexasHold'em poker (Brown & Sandholm, 2019), and AlphaStar for the game Starcraft (Vinyals et al., 2019). In order to tackle problems with large state/action spaces, researchers have designed MARL algorithms based on *function approximation* which approximate the original high-dimensional value function/policy by a function approximator. For instance, Xie et al. (2020); Chen et al. (2021) studied RL for two-player MGs with *linear function approximation*, where it is assumed that there are a set of *linear features* that span the transition kernel and reward function spaces. In contrast to RL with linear function approximation, RL with *nonlinear function approximation* (e.g. neural network approximation) aims to take advantage of the superior representational power of nonlinear function compared to linear parameterizations. For example, Jin et al. (2021) studied neural-network-based RL in the setting of *MGs with low multi-agent Bellman eluder dimension*, obtaining algorithms that have polynomial dependence on the complexity of the underlying function class. Although this yields a strong theoretical guarantee, the specific algorithm that they propose is not computationally efficient due to the constructed highly nonconvex confidence sets. The following question is still open:

Can we design a computationally efficient and sample efficient RL algorithm for learning two-player Markov games with nonlinear functions with RKHS as the function classes?

In this paper, we give an affirmative answer to this question for a class of episodic Markov games. We propose the `KernelCCE-VTR`, a novel RKHS-based (Reproducing Kernel Hilbert Space) MARL algorithmic framework for general episodic two-player MGs that provides provable regret guarantees. We summarize the contributions of our work as follows:

- We propose a specific `KernelCCE-VTR` algorithm for two-player MGs. In particular, at each episode, `KernelCCE-VTR` uses an RKHS function approximation (for instance, overparameterized two-layer neural network) to approximate the optimal value function and constructs corresponding confidence sets, following the "Optimism-in-Face-of-Uncertainty" principle (Abbasi-Yadkori et al., 2011) to selection an action based on current state. In contrast to algorithms in Jin et al. (2021), which construct implicit confidence sets that are generally intractable computationally, our `KernelCCE-VTR` crafts computationally efficient exploration bonus terms based directly on the gradient features induced by the neural networks.
- We show that under a closure assumption on the optimal value function class, our `KernelCCE-VTR` is able to find a Nash equilibrium of the game with a $\widetilde{O}(d_K H^2\sqrt{T})$ upper bound on the duality gap, where H is the horizon, T is the number of the episodes, and d_K is determined by the effective dimension and is an indicator of the complexity of the function class $\mathcal{F}$. When $\mathcal{F}$ reduces to the linear function class, our result reduces to $\widetilde{O}(dH^2\sqrt{T})$ and nearly matches the complexity result in Chen et al. (2021) up to a $\sqrt{H}$ factor. To the best of our knowledge, this is the first algorithm for learning two-player Markov Games with nonlinear function function approximation that is efficient in terms of both computational and sample complexities.

Notations We use lower case letters to denote scalars, lower and upper case bold letters to denote vectors and matrices. We use $\|\cdot\|$ to indicate Euclidean norm, and for a semi-positive definite matrix $\mathbf{\Sigma}$ and any vector $\mathbf{x}$, $\|\mathbf{x}\|_{\mathbf{\Sigma}} := \|\mathbf{\Sigma}^{1/2}\mathbf{x}\| = \sqrt{\mathbf{x}^\top\mathbf{\Sigma}\mathbf{x}}$. For a real value x and an interval $[a, b]$, we use $[x]_{[a,b]}$ to indicate the projection of x onto $[a, b]$. We also use the standard O and Ω notations. We say $a_n = O(b_n)$ if and only if there exists $C > 0, N > 0$, for any $n > N$, $a_n \leq Cb_n$; $a_n = \Omega(b_n)$ if $a_n \geq Cb_n$. The notation $\widetilde{O}$ is used to hide logarithmic factors.

2 Related Works

Online RL with function approximation MARL with function approximation can be seen as an extension of RL with function approximation on MDPs. There are several lines of works studying RL with function approximation. The first line of works study the so-called linear MDP which assumes the reward function and transition dynamics are linear functions of some feature mapping defined on state and action spaces (Yang & Wang, 2020; Jin et al., 2020; Zanette et al., 2020). These works proposed model-free algorithms with sublinear regrets on the number of episodes K. The second line of works study the linear mixture MDP which assumes the transition kernel is a linear combination of several base models (Modi et al., 2020; Jia et al., 2020; Zhou et al., 2020a; 2021). These works proposed model-based RL algorithms that estimate the transition kernel with finite sample complexity or sublinear regret guarantees. The third line of works study the general function approximation which assumes either the value function or the transition kernel can be approximated by a general class of functions (Osband & Van Roy, 2014; Jiang et al., 2017; Sun et al., 2019; Wang et al., 2020; Yang et al., 2020). Algorithms proposed in this line enjoy finite regret or sample complexity bounds that depend on some general complexity measures such as Eluder dimension (Russo & Van Roy, 2013; Osband & Van Roy, 2014), Bellman rank (Jiang et al., 2017), witness rank (Sun et al., 2019), and information gain (Yang et al., 2020).

Learning two-player MGs with function approximation There is a large body of literature on MARL for two-player MGs with function approximation. These works can be generally categorized into MARL with *linear function approximation* and MARL with *general function approximation*. To mention a few, for linear function approximation, Xie et al. (2020) studied the zero-sum simultaneous-move MGs on the linear MGs where both the reward and transition kernel can be parameterized as linear functions of some feature mappings. They proposed an OMVI-NI algorithm with an $\widetilde{O}(\sqrt{d^3H^3T})$ regret, where d is the number of the feature dimension, H is the episode length and T is the total number of rounds. Chen et al. (2021) studied the linear mixture MGs and proposed a nearly minimax optimal Nash-UCRL-VTR algorithm with an $\widetilde{O}(dH\sqrt{T})$ regret and an $\Omega(dH\sqrt{T})$

lower bound of regret. In contrast to these two works, our `KernelCCE-VTR` does not assume the underlying transition dynamic or reward function have a linear structure. For MARL with general function approximation, Jin et al. (2021) studied the two-player zero-sum MGs with low multi-agent Bellman-Eluder dimension and proposed a Golf with Exploiter algorithm using a general function class. They showed their algorithm enjoys an $\widetilde{O}(H\sqrt{dK\log N})$ regret, where d is the multi-agent Bellman-Eluder dimension, K is the number of episodes.Huang et al. (2021) studied two-player MGs with a finite minimax Eluder dimension and proposed an ONEMG method with an $\widetilde{O}(H\sqrt{dK\log N})$ regret, where d is the minimax Eluder dimension. To obtain the desired function approximator, both Golf with Exploiter and ONEMG need to solve a constrained optimization problem, and such a solution is computational intractable even under the linear function approximation setting. In contrast to Jin et al. (2021); Huang et al. (2021), our proposed `KernelCCE-VTR` is computationally efficient.

3 PRELIMINARIES

In this section, we present the definition of a two-player Markov game. We define the action value function corresponding to two player's policies and introduce the Nash-Equilibrium solution where both players' policy act as each other's best response policy. As a slightly relaxed version of the Nash-Equilibrium (NE), we also introduce the Coarse Correlated Equilibrium (CCE), which serves as a computational efficient approximation of NE. Finally, we introduce the neural network function class in an overparameterized regime. We define the Neural Tangent Kernel (NTK) K_{ntk} corresponding to a neural network and introduce the Reproducing Kernel Hilbert Space (RKHS) function classes$\mathcal{H}$ using the neural tangent kernel.

3.1 TWO-PLAYER MARKOV GAMES

In this subsection, for two-player Markov games, we describe the settings of a simultaneous move games. A simpler instance of Markov games named turn-based games can be seen as a special case of simultaneous move games. We present the introduction and results of turn-based games in later sections.

In a two player simultaneous move Markov game, the dynamical structure can be captured by an MG formulated as $(\mathcal{S}, \mathcal{A}_1, \mathcal{A}_2, r, \mathbb{P}, H)$. $\mathcal{S}$ is the space of available states of the environment. $\mathcal{A}_1$ is the action space of the first player and $\mathcal{A}_2$ is the action space of the second player. H is the time horizon representing the maximum step of each round of play. The reward function $r : \{r_h(x, a, b) : h \in [H]\}$ is a sequence of mappings from $\mathcal{S} \times \mathcal{A}_1 \times \mathcal{A}_2$ to $[-1, 1]$. And the transition matrix $\mathbb{P} : \{\mathbb{P}_h(\cdot \mid x, a, b) : h \in [H]\}$ gives for each state actions triplet (x, a, b) and at each time h the stochastic response of the environment to the next $x' \in \mathcal{S}$. Here by simultaneous move we refer to the setting where at each round of game the two players P_1 and P_2 take actions $a \in \mathcal{A}_1, b \in \mathcal{A}_2$ simultaneously at a given state $x \in \mathcal{S}$, in contrast with the turn-based game where r_h and $\mathbb{P}_h$ are defined for a state-action pair (x, a) where the action can be taken by either players. In the context of this paper, for simplicity of notations we let $\mathcal{A}_1 = \mathcal{A}_2$, while the results can be easily generalized to the case when $\mathcal{A}_1 \neq \mathcal{A}_2$. Similar definitions of a two player simultaneous move episodic Markov games can be found in Xie et al. (2020); Perolat et al. (2018); Wei et al. (2017).

Under the above settings, two players P_1 and P_2 take actions according to their own strategies. We use $\pi := \{\pi_h\}_{h\in[H]}$ to denote the stochastic policy of P_1 and use $\nu := \{\nu_h\}_{h\in[H]}$ to denote the stochastic policy of P_2. We note that at time h, $\pi_h : \mathcal{S} \mapsto \Delta_{\mathcal{A}}$ maps the current state x_s to a probability distribution of the actions. As is the same with ν_h. Given two agents policies π, ν across h steps, the state value function is defined as the expected total reward through H steps when at step $h \in [H]$ player P_1 follows policy $\pi_h(\cdot \mid x_h)$ and player P_2 follows policy $\nu_h(\cdot \mid x_h)$. Note that the expectation is taken over all stochasticity in π_h, ν_h and $\mathbb{P}_h$.

$$V_h^{\pi,\nu}(x) := \mathbb{E}_{\pi,\nu}\left[\sum_{t=h}^{H} r_t(s_t, a_t, b_t) \mid x_h = x\right], \quad \text{and} \quad V^{\pi,\nu}(x) := V_1^{\pi,\nu}(x).$$

The action value function is defined as

$$Q_h^{\pi,\nu}(x, a, b) := \mathbb{E}_{\pi,\nu}\left[\sum_{t=h}^{H} r_t(x_t, a_t, b_t) \mid x_h = x, a_h = a, b_h = b\right], \quad \text{and} \quad Q^{\pi,\nu}(x, a, b) := Q_1^{\pi,\nu}$$

From the definition of two value functions, we observe that for any $x \in \mathcal{S}$, the state value function given policy pair (π, ν) is the expectation of the corresponding action value function where the expectation is taken over the actions distribution induced by the policy pair:

$$V_h^{\pi,\nu}(x) = \mathbb{E}_{(a,b)\sim(\pi,\nu)} Q_h^{\pi,\nu}(x, a, b).$$

3.2 Equilibrium and Duality Gap

Nash Equilibrium and Duality Gap In a two player Markov game, P_1 wants to maximize the expected reward $V^{\pi,\nu}(x)$ by properly choosing its policy π. On the contrary, P_2 wants to minimize $V^{\pi,\nu}(x)$ by properly choosing ν. For fixed ν, we define the best response policy with respect to V and ν as $\text{br}(\nu)$ and define $V_h^{*,\nu} := V_h^{\text{br}(\nu),\nu}$ and $Q_h^{*,\nu} := Q_h^{\text{br}(\nu),\nu}$, similarly we define $V_h^{\pi,*} := V_h^{\pi,\text{br}(\pi)}$ and $Q_h^{\pi,*} := Q_h^{\pi,\text{br}(\pi)}$. The Nash Equilibrium is a pair of policies (π^*, ν^*) that are the best response policy of each other, meaning

$$V^{\pi^*,*}(x) = V^{\pi^*,\nu^*}(x) = V^{*,\nu^*}(x).$$

For notational simplicity we write $V^* := V^{\pi^*,\nu^*}, Q^* := Q^{\pi^*,\nu^*}$. By definition of the best response policy, we have the following weak duality

$$V_h^{\pi,*}(x) \leq V_h^*(x) \leq V_h^{*,\nu}(x).$$

We define the duality gap as $\sum_{t=1}^T V_1^{*,\nu^t}(x_1^t) - V_1^{\pi^t,*}(x_1^t)$ and use it as the regret in the Markov game setting. This metric has been widely used in previous literature.

Coarse Correlated Equilibrium Given the estimates of the value function corresponding to P_1 and P_2 respectively, the problem of finding an optimal policy reduces to finding the Nash Equilibrium of a general sum game. For computational efficiency, we introduce the Coarse Correlated Equilibrium (CCE) as a substitute of the NE solution. Given payoff matrices $Q_1, Q_2 : \mathcal{S} \times \mathcal{A} \times \mathcal{A} \mapsto \mathbb{R}$, we define the CCE of the game as a joint distribution σ on $\mathcal{A} \times \mathcal{A}$ satisfying:

$$\mathbb{E}_{(a,b)\sim\sigma}\left[Q_1(x, a, b)\right] \geq \mathbb{E}_{b\sim\mathcal{P}_2\sigma}\left[Q_1\left(x, a', b\right)\right], \quad \forall a' \in \mathcal{A},$$
$$\mathbb{E}_{(a,b)\sim\sigma}\left[Q_2(x, a, b)\right] \leq \mathbb{E}_{a\sim\mathcal{P}_1\sigma}\left[Q_2\left(x, a, b'\right)\right], \quad \forall b' \in \mathcal{A},$$

where $\mathcal{P}_1\sigma$ denotes the marginal of σ on the first coordinate and $\mathcal{P}_2\sigma$ denotes the marginal of σ on the second coordinate.

3.3 Nonlinear Function Approximation by Reproducing Kernel Hilbert Spaces

In this subsection, we provide necessary definitions and notations in approximating Q with functions belonging to an reproducing kernel hilbert spaces (RKHS) via modeling the transition probability. For simplicity of notations, we use $z = (x, a, b)$ to denote the state action triplet in $\mathcal{Z} := \mathcal{S} \times \mathcal{A} \times \mathcal{A}$.

An RKHS $\mathcal{H}$ with kernel $K(\cdot,\cdot) : \mathcal{Z} \times \mathcal{Z} \mapsto \mathbb{R}$ is a generalization of the linear function class. Every RKHS $\mathcal{H}$ consists of functions on $\mathcal{Z}$, where there exists a feature mapping $\phi : \mathcal{Z} \mapsto \mathcal{H}$, such that $\forall f \in \mathcal{H}$ and $\forall z \in \mathcal{Z}$, $f(z) = \langle f, \phi(z)\rangle_{\mathcal{H}}$. The kernel K is thus defined for every $x, y \in \mathcal{Z} \times \mathcal{Z}$ as $K(x, y) = \langle \phi(x), \phi(y)\rangle_{\mathcal{H}}$. We call ϕ the feature mapping induced by the RKHS $\mathcal{H}$ with kernel K. In the following sections, we use $f^\top g$ as a simplification of $\langle f, g\rangle_{\mathcal{H}}$ when $f, g \in \mathcal{H}$. We make no distinction in notations between the vector product and the product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$, however, one can tell the difference from the two objects of the product. For every RKHS $\mathcal{H}$, there exists a natural eigenvalue decomposition in $\mathcal{L}^2(\mathcal{Z})$. RKHS approximation is a generalization of the linear function approximation of finite dimension d, which can be infinite dimensional. In the following, we define the so-called *kernel mixture MG*, which can be regarded as an extension from the linear mixture MDP (Jia et al., 2020; Ayoub et al., 2020; Zhou et al., 2021) and linear mixture MG (Chen et al., 2021) to their kernel counterpart.

Kernel Mixture MG In a kernel mixture MG model, we model the transition probability $\mathbb{P}_h(s' \mid z) : \mathcal{Z} \mapsto \Delta(\mathcal{S})$ through an RKHS product of a feature mapping $\phi(s' \mid z) : \mathcal{Z} \mapsto \mathcal{S} \times \mathcal{H}$ with an unknown parameter $\boldsymbol{\theta}_h^* \in \mathcal{H}$:

$$\mathbb{P}_h(s' \mid z) = \langle \phi(s' \mid z), \boldsymbol{\theta}_h^*\rangle_{\mathcal{H}}, \quad \forall s' \in \mathcal{S}.$$

At time h, for any estimate of the value function $V_h(\cdot) : \mathcal{S} \mapsto \mathbb{R}$, we note that the expected value function at time $h+1$, $\mathbb{P}_h V_{h+1}$ is an inner product in the RKHS space

$$\mathbb{P}_h V_{h+1}(z) = \left\langle \phi_{V_{h+1}}(z), \boldsymbol{\theta}_h^* \right\rangle_{\mathcal{H}},$$

where $\phi_{V_{h+1}}(z) := \sum_{s' \in \mathcal{S}} \phi(s' \mid z) V_{h+1}(s')$, integrates the product of the feature map with the estimated value of s' on $\mathcal{S}$. During this work, we need to access the inner product function $k_{V_1,V_2}(\cdot,\cdot)$, which is defined as follows:

$$k_{V_1,V_2}(z_1,z_2) := \sum_{s_1,s_2 \in \mathcal{S}} V_1(s_1)V_2(s_2) \cdot \langle \phi(s_1|z_1), \phi(s_2|z_2) \rangle_{\mathcal{H}}, \ \forall V_1, V_2 : \mathcal{S} \to [0,1], \ z_1, z_2 \in \mathcal{Z}.$$

It is easy to see that $k_{V_1,V_2}(z_1,z_2) = \langle \phi_{V_1}(z_1), \phi_{V_2}(z_2) \rangle_{\mathcal{H}}$. During this work, we assume that we can access an integration oracle that can calculate $K_{V_1,V_2}(z_1,z_2)$ for any function V_1, V_2 and state-action tuples z_1, z_2 efficiently. We also assume that for any bounded value function $V(\cdot) : \mathcal{S} \mapsto [-1,1]$ and any $z \in \mathcal{Z}$, $||\phi_V(z)|| \leq 1$. Given that the reward function $r_h(z)$ is known, we obtain through the Bellman equation that

$$Q_h^{*,\nu}(\cdot) = r_h(\cdot) + (\mathbb{P}_h V_{h+1}^{*,\nu})(\cdot) = r_h(\cdot) + \left\langle \phi_{V_{h+1}^{*,\nu}}(\cdot), \boldsymbol{\theta}_h^* \right\rangle_{\mathcal{H}},$$
$$Q_h^{\pi,*}(\cdot) = r_h(\cdot) + (\mathbb{P}_h V_{h+1}^{\pi,*})(\cdot) = r_h(\cdot) + \left\langle \phi_{V_{h+1}^{\pi,*}}(\cdot), \boldsymbol{\theta}_h^* \right\rangle_{\mathcal{H}}.$$

With these preliminary settings prepared, we now move to our main algorithm and its analysis.

4 Algorithm

In this section, we introduce our value targeted iteration algorithm for the two player Markov Game setting under RKHS function approximation. We follow the "Value-targeted regression" framework and the confidence set design as in UCRL (Jia et al., 2020; Ayoub et al., 2020), combine the CCE techniques (Xie et al., 2020) to deal with the general sum sub-game brought by seperate UCB and LCB value functions. These techniques enables us to adapt the results in the linear setting to the nonlinear RKHS regime (Yang et al., 2020; Zhou et al., 2020b; Chowdhury & Gopalan, 2017) to get a structure dependent regret bound that is both computationally simple and statistically efficient.

Aiming at finding an equilibrium (π^*, ν^*) of the value function $V_1^{\pi,\nu}(x_1)$, we design an algorithm based on value targeted regression (VTR) and upper/lower confident estimations. As the min-player targets at minimizing the value function while the max-player targets at maximizing the value function, we use an upper confident estimation to encourage exploration of the max-player and use a lower confidence estimation to encourage exploration of the min-player. Thus we need to define separately two value functions for the min-max-players, we denote the value functions as $\overline{Q}_h^t, \underline{Q}_h^t, \overline{V}_h^t, \underline{V}_h^t$, where we adopt the overline notation for the over estimation of the max-player and the underline notation for the under estimation of the min-player. At each round of play, we solve the following ridge regression problem for minimizing the Bellman error.

$$\begin{aligned}
\overline{\boldsymbol{\theta}}_h^t &= \min_{\boldsymbol{\theta} \in \mathcal{H}} \sum_{\tau=1}^{t-1} \left[\overline{V}_{h+1}^{\tau}(x_{h+1}^{\tau}) - \left\langle \phi_{\overline{V}_{h+1}^{\tau}}(z_h^{\tau}), \boldsymbol{\theta} \right\rangle_{\mathcal{H}} \right]^2 + \lambda \cdot ||\boldsymbol{\theta}||_{\mathcal{H}}^2 \\
\underline{\boldsymbol{\theta}}_h^t &= \min_{\boldsymbol{\theta} \in \mathcal{H}} \sum_{\tau=1}^{t-1} \left[\underline{V}_{h+1}^{\tau}(x_{h+1}^{\tau}) - \left\langle \phi_{\underline{V}_{h+1}^{\tau}}(z_h^{\tau}), \boldsymbol{\theta} \right\rangle_{\mathcal{H}} \right]^2 + \lambda \cdot ||\boldsymbol{\theta}||_{\mathcal{H}}^2
\end{aligned} \tag{4.1}$$

We note that in Eq. (4.1), each $\overline{V}_{h+1}^{\tau}, \underline{V}_{h+1}^{\tau}$ only depends on the previous trajectories $\left\{ x_i^j, a_i^j, b_i^j : j \in [\tau-1], i \in [H] \right\}$. We denote the corresponding σ-algebra as $\mathcal{F}_{\tau-1}$. Thus we have $\overline{V}_{h+1}^{\tau}, \underline{V}_{h+1}^{\tau} \in \mathcal{F}_{\tau-1}$. As each of the $\overline{V}_{h+1}^{\tau}(x_{h+1}^{\tau})$ can be seen as a stochastic sample of $(\mathbb{P}_h \overline{V}_{h+1}^{\tau})(z_h^{\tau})$. The regularized regression problem of the max-player in (4.1) can be seen as a bandit problem with context $\phi_{\overline{V}_{h+1}^{\tau}}(z_h^{\tau})$, reward function $(\mathbb{P}_h \overline{V}_{h+1}^{\tau})(z_h^{\tau})$ and noise term $\overline{V}_{h+1}^{\tau}(x_{h+1}^{\tau}) - (\mathbb{P}_h \overline{V}_{h+1}^{\tau})(z_h^{\tau})$. Similar statement holds for the min-player as well.

From the solution of the ridge regression problem (4.1), we are to define the upper/lower confidence estimate of the value functions $Q_h^{*,\nu}, Q_h^{\pi,*}$ respectively. For simplicity of notations we define the vectors

$\overline{\mathbf{\Psi}}_h^t := \left(\phi_{\overline{V}_{h+1}^1}(z_h^1), \ldots \phi_{\overline{V}_{h+1}^{t-1}}(z_h^{t-1})\right)^\top \in \mathcal{H}^{t-1}$ and $\underline{\mathbf{\Psi}}_h^t := \left(\phi_{\underline{V}_{h+1}^1}(z_h^1), \ldots \phi_{\underline{V}_{h+1}^{t-1}}(z_h^{t-1})\right)^\top \in \mathcal{H}^{t-1}$. Also, we define the gram matrix $\overline{K}_h^t$ and vector-valued function $\overline{k}_h^t$ as

$$\overline{K}_h^t = \left(\overline{\mathbf{\Psi}}_h^t\right)\left(\overline{\mathbf{\Psi}}_h^t\right)^\top \in \mathbb{R}^{(t-1)\times(t-1)},$$
$$\overline{k}_h^t = \left(\overline{\mathbf{\Psi}}_h^t\right)\phi_{\overline{V}_{h+1}^t}(z) = \left(k_{\overline{V}_{h+1}^i, \overline{V}_{h+1}^t}(z_h^i, z)\right)_i \in \mathbb{R}^{t-1}.$$

We define $\underline{K}_h^t$ and $\underline{k}_h^t$ in a similar way. For a positive parameter $\beta_t > 0$ that will be chosen in later analysis, the confidence region centered at $\overline{\boldsymbol{\theta}}_h^t, \underline{\boldsymbol{\theta}}_h^t$ in the RKHS space $\mathcal{H}$ is defined as

$$\overline{\mathcal{C}}_h^t = \left\{\boldsymbol{\theta} : \sqrt{\lambda||\boldsymbol{\theta} - \overline{\boldsymbol{\theta}}_h^t||_{\mathcal{H}}^2 + \left|\left\langle \overline{\mathbf{\Psi}}_h^t, \boldsymbol{\theta} - \overline{\boldsymbol{\theta}}_h^t\right\rangle_{\mathcal{H}}\right|^2} \leq \beta_t\right\},$$

where we use $\left\langle \overline{\mathbf{\Psi}}_h^t, \boldsymbol{\theta} - \boldsymbol{\theta}_h^t\right\rangle_{\mathcal{H}}$ to denote the following $t-1$ dimensional vector:

$$\left(\left\langle \phi_{\overline{V}_{h+1}^\tau}(z_h^\tau), \boldsymbol{\theta} - \boldsymbol{\theta}_h^t\right\rangle_{\mathcal{H}} : 1 \leq \tau \leq t-1\right).$$

We omit the definition of $\underline{\mathcal{C}}_h^t$ which is an analogue of Eq. (4.2) by changing all overline symbols to underline. Based on the confidence region, We construct an optimistic/pessimistic estimate of $Q_h^{*,\nu}$ as

$$\overline{Q}_h^t := \left[r_h + \max_{\boldsymbol{\theta} \in \overline{\mathcal{C}}_h^t} \left\langle \phi_{\overline{V}_{h+1}^t}, \boldsymbol{\theta}\right\rangle_{\mathcal{H}}\right]_{[-H,H]}, \quad \underline{Q}_h^t := \left[r_h + \min_{\boldsymbol{\theta} \in \underline{\mathcal{C}}_h^t} \left\langle \phi_{\underline{V}_{h+1}^t}, \boldsymbol{\theta}\right\rangle_{\mathcal{H}}\right]_{[-H,H]}.$$

In fact, $\overline{Q}_h^t$ has a closed form solution as follows:

$$\overline{Q}_h^t(z) = \left[r_h(z) + \left\langle \phi_{\overline{V}_{h+1}^t}(z), \overline{\boldsymbol{\theta}}_h^t\right\rangle + \beta_t \cdot \overline{b}_h^t(z)\right]_{[-H,H]}, \tag{4.2}$$

where

$$\overline{b}_h^t(z) = \lambda^{-1/2} \cdot \left[k_{\overline{V}_{h+1}^t, \overline{V}_{h+1}^t}(z,z) - \overline{k}_h^t(z)^\top \left(\overline{K}_h^t + \lambda \cdot \mathbf{I}\right)^{-1} \overline{k}_h^t(z)\right]^{1/2}.$$

Same claim also holds for $\underline{Q}_h^t$. Given the estimation of $\overline{Q}_h^t, \underline{Q}_h^t$, the next step is to estimate the corresponding state value functions $\overline{V}_h^t, \underline{V}_h^t$. Due to the computational difficulty of calculating the NE of a general sum game (Daskalakis et al., 2009), we use a more efficient solution of finding a CCE of the payoff pair $(\overline{Q}_h^t(z), \underline{Q}_h^t(z))$. We utilize the `FIND_CCE` algorithm in Xie et al. (2020). The full version of the algorithm is presented formally as in Algorithm 1.

5 Theoretical Analysis

For linear function class, the regret upper bound is characterized by the dimension of the linear function, the horizon of the game, and the number of episodes. When considering the nonlinear function class as approximator of the value function, we need to develop a new concepts analogue to the dimension d that characterize the intrinsic complexity of the function class $\mathcal{F}$. Under the framework of our theoretical analysis, we present our regret bound in terms of the maximal information gain $\Gamma_K(T, \lambda)$ (Srinivas et al., 2009), the episode number T, and the time horizon H. We nail down precise definitions of these notions immediately afterwards.

5.1 Main Result on RKHS

In this subsection, we show that our algorithm in the RKHS setting reduces to the linear function approximation when $K(z, z') = \phi(z)^\top \phi(z')$ and obtains a regret bound of $\widetilde{\mathcal{O}}(dH^2\sqrt{T})$. We first define the *effective dimension* of the RKHS $\mathcal{H}$ with respect to the kernel mixture MG as follows:

Algorithm 1 KernelCCE-VTR

Input: bonus parameter $\beta > 0$.
for episode $t = 1, 2, \ldots, T$ **do**
  Receive initial state x_1^t
  for step $h = H, H-1, \ldots, 1$ **do**
    Calculate $\overline{Q}_h^t(\cdot), \underline{Q}_h^t(\cdot)$ as in Eq. (4.2)
    For each x, let $\sigma_h^t(x) = \texttt{FIND_CCE}(\overline{Q}_h^t, \underline{Q}_h^t, x)$
    Let $\overline{V}_h^t(x) = \mathbb{E}_{(a,b)\sim\sigma_h^t(x)}\overline{Q}_h^t(x,a,b)$ and $\underline{V}_h^t(x) = \mathbb{E}_{(a,b)\sim\sigma_h^t(x)}\underline{Q}_h^t(x,a,b)$
  end for
  for step $h = 1, 2, \ldots, T$ **do**
    Sample $(a_h^t, b_h^t) \sim \sigma_h^t(x_h^t)$.
    P_1 takes action a_h^t, P_2 takes action b_h^t
    Observe next state x_{h+1}^t.
  end for
end for

Definition 1. We define the effective dimension $\Gamma_K(T, \lambda)$ as follows:

$$\Gamma_K(T,\lambda) := \sup_{(V_i)_i,(z_i)_i} \frac{1}{2}\log\det(\mathbf{I} + K((V_i)_i,(z_i)_i)/\lambda),\ \forall 1 \le i \le T,\ V_i : \mathcal{S} \to [0,1],\ z_i \in \mathcal{Z},$$

where $K((V_i)_i,(z_i)_i) \in \mathbb{R}^{T\times T}$ and its (p,q)-th entry is

$$[K((V_i)_i,(z_i)_i)]_{p,q} = k_{V_p,V_q}(z_p,z_q), 1 \le p,q \le T.$$

It is easy to verify that both the tabular MG and the linear mixture MG enjoy a finite effective dimension. Via a union concentration bound argument, we first present our main Lemma 2 for bounding the estimation error:

Lemma 2. Assuming that for any $h \in [H]$, $||\boldsymbol{\theta}_h^*||_{\mathcal{H}} \le B$. Let $\lambda = 1 + \frac{1}{T}$ and β satisfies

$$\left(\frac{\beta}{H}\right)^2 \ge 2\Gamma_K(T,\lambda) + 2 + 4\cdot\log\left(\frac{1}{\delta}\right) + 2\lambda\left(\frac{B}{H}\right)^2.$$

Then for any $\delta > 0$, with probability at least $1-\delta$ the following holds for any $(t,h) \in [T]\times[H]$ and any $(x,a) \in \mathcal{S}\times\mathcal{A}$:

$$\left|\left\langle \phi_{\overline{V}_{h+1}^t}, \overline{\boldsymbol{\theta}}_h^t - \boldsymbol{\theta}_h^* \right\rangle_{\mathcal{H}}\right| \le \beta\overline{b}_h^t(x,a,b),$$
$$\left|\left\langle \phi_{\underline{V}_{h+1}^t}, \overline{\boldsymbol{\theta}}_h^t - \boldsymbol{\theta}_h^* \right\rangle_{\mathcal{H}}\right| \le \beta\underline{b}_h^t(x,a,b).$$

We are now ready to present the main theorem as follows:

Theorem 3 (Main Results on RKHS)**.** Assuming that for any $h \in [H]$, $||\boldsymbol{\theta}_h^*||_{\mathcal{H}} \le B$. Set $\lambda = 1 + \frac{1}{T}$ in the `KernelCCE-VTR` Algorithm. For any $\delta > 0$ and any β satisfying

$$\left(\frac{\beta}{H}\right)^2 \ge 2\Gamma_K(T,\lambda) + 2 + 4\cdot\log\left(\frac{1}{\delta}\right) + 2\lambda\left(\frac{B}{H}\right)^2,$$

with probability at least $1-\delta$, we have

$$\text{Regret}(T) \le 5\beta H\sqrt{T\cdot\Gamma_K(T,\lambda)} + 8.$$

Theorem 3 provides the $O(\sqrt{T})$ regret bound, and β has the practical meaning as the *radius* of the confidence region. It is also worth-mentioning that when the RKHS degenerates to the Euclidean case, the regret bound corresponding to the standard linear mixture case simply reduces to $dH^2\sqrt{T}$ which is known to be nearly minimax optimal up to a $\sqrt{H}$ factor (Zhou et al., 2020a). However, an extra integration oracle is required to be accessible to achieve such minimax rate.

5.2 SKETCH OF PROOF

In this subsection, we sketch the proof of our main Theorem 3 on RKHS. We recall that the duality gap is defined as $\sum_{t=1}^{T} V_1^{*,\nu^t}(x_1^t) - V_1^{\pi^t,*}(x_1^t)$. As can be seen in our Algorithm, we maintain an optimistic estimate of $V_h^{*,v^t}(\cdot)$ as $\overline{V}_{h+1}^t(\cdot)$ and a pessimistic estimate of $V_h^{\pi^t,*}(\cdot)$ as $\underline{V}_{h+1}^t(\cdot)$. Hence the term $\overline{V}_{h+1}^t(x_h^t) - \underline{V}_{h+1}^t(x_h^t)$ is approximately the upper bound of the duality gap. We write the decomposition formally as below:

$$V_h^{*,\nu^t}(x_h^t) - V_h^{\pi^t,*}(x_h^t) = \underbrace{\overline{V}_h^t(x_h^t) - \underline{V}_h^t(x_h^t)}_{\text{I}} - \underbrace{\left(V_h^{\pi^t,*}(x_h^t) - \underline{V}_h^t(x_h^t)\right)}_{\text{II}} - \underbrace{\left(\overline{V}_h^t(x_h^t) - V_h^{*,\nu^t}(x_h^t)\right)}_{\text{III}}. \tag{5.1}$$

We use $\overline{\delta}_h^t$ to denote an important quantity $\left\langle \phi_{\overline{V}_{h+1}^t}(z_h^t), \boldsymbol{\theta}_h^* - \overline{\boldsymbol{\theta}}_h^t \right\rangle_{\mathcal{H}}$ (and $\left\langle \phi_{\underline{V}_{h+1}^t}(z_h^t), \boldsymbol{\theta}_h^* - \underline{\boldsymbol{\theta}}_h^t \right\rangle_{\mathcal{H}}$) in estimating the duality gap. In the rest of the proof we aim to show that all of the above three terms can be bounded by a quantity related to $\overline{\delta}_h^t$ ($\underline{\delta}_h^t$) and a stochastic random variable that forms a martingale difference sequence when considering for all $h \in [H], t \in [T]$.

For bounding term I, we first define two sequences of zero mean variables:

$$\begin{aligned} \gamma_h^t &:= \overline{Q}_h^t(x_h^t, a_h^t, b_h^t) - \underline{Q}_h^t(x_h^t, a_h^t, b_h^t) - \mathbb{E}_{(a,b)}\left[\overline{Q}_h^t(x,a,b) - \underline{Q}_h^t(x,a,b)\right], \\ \xi_h^t &:= \left(\mathbb{P}_h(\overline{V}_{h+1}^t - \underline{V}_{h+1}^t)\right)(x_h^t, a_h^t, b_h^t) - \left(\overline{V}_{h+1}^t(x_{h+1}^t) - \underline{V}_{h+1}^t(x_{h+1}^t)\right), \end{aligned} \tag{5.2}$$

where γ_h^t depicts the stochastic error with respect to the policy and ξ_h^t depicts the stochastic error with respect to the transition. We refer the readers to the proof of Lemma 4 for detailed explanations on these two error term. Given the above definition, we have the following Lemma 4.

Lemma 4. Under the settings of Lemma 2, we have the following recursive bound:

$$\overline{V}_h^t(x_h^t) - \underline{V}_h^t(x_h^t) \le \overline{V}_{h+1}^t(x_{h+1}^t) - \underline{V}_{h+1}^t(x_{h+1}^t) + 4\beta b_h^t(x_h^t) + \xi_h^t + \gamma_h^t. \tag{5.3}$$

Proof of Lemma 4. By the update rule of Algorithm 1, we have the following relation:

$$\overline{V}_h^t(x_h^t) - \underline{V}_h^t(x_h^t) = \mathbb{E}_{(a,b)\sim\sigma_h^t(x_h^t)}\left[\overline{Q}_h^t(x_h^t, a, b) - \underline{Q}_h^t(x_h^t, a, b)\right]. \tag{5.4}$$

We note that the RHS of Eq. (5.4) is an expectation over the CCE equilibrium distribution $\sigma_h^t(x_h^t)$, which can be decomposed into one sample from the distribution plus a noise term as follows:

$$\overline{V}_h^t(x_h^t) - \underline{V}_h^t(x_h^t) = \overline{Q}_h^t(x_h^t, a_h^t, b_h^t) - \underline{Q}_h^t(x_h^t, a_h^t, b_h^t) + \gamma_h^t, \tag{5.5}$$

where $\gamma_h^t := \overline{Q}_h^t(x_h^t, a_h^t, b_h^t) - \underline{Q}_h^t(x_h^t, a_h^t, b_h^t) - \mathbb{E}_{(a,b)}\left[\overline{Q}_h^t(x,a,b) - \underline{Q}_h^t(x,a,b)\right]$. Furthermore, by applying Lemma 2, we have

$$\begin{aligned} \overline{Q}_h^t(z_h^t) - \underline{Q}_h^t(z_h^t) &\le \left\langle \phi_{\overline{V}_{h+1}^t}(z_h^t), \overline{\boldsymbol{\theta}}_h^t \right\rangle_{\mathcal{H}} - \left\langle \phi_{\underline{V}_{h+1}^t}(z_h^t), \underline{\boldsymbol{\theta}}_h^t \right\rangle_{\mathcal{H}} + \beta\overline{b}_h^t(z_h^t) - \beta\underline{b}_h^t(z_h^t) \\ &\le \left(\mathbb{P}_h(\overline{V}_{h+1}^t - \underline{V}_{h+1}^t)\right)(x_h^t, a_h^t, b_h^t) + 2\beta\overline{b}_h^t(z_h^t) + 2\beta\underline{b}_h^t(z_h^t), \end{aligned}$$

where again by extracting the sequence

$$\xi_h^t := \left(\mathbb{P}_h(\overline{V}_{h+1}^t - \underline{V}_{h+1}^t)\right)(x_h^t, a_h^t, b_h^t) - \left(\overline{V}_{h+1}^t(x_{h+1}^t) - \underline{V}_{h+1}^t(x_{h+1}^t)\right),$$

we have

$$\overline{Q}_h^t(x_h^t, a_h^t, b_h^t) - \underline{Q}_h^t(x_h^t, a_h^t, b_h^t) \le \overline{V}_{h+1}^t(x_{h+1}^t) - \underline{V}_{h+1}^t(x_{h+1}^t) + 2\beta\overline{b}_h^t(z_h^t) + 2\beta\underline{b}_h^t(z_h^t) + \xi_h^t. \tag{5.6}$$

Combining Eq. (5.5) and (5.6) concludes our proof. □

For bounding II and III, we observe that the gap between two consecutive points of the values of term II can be decomposed as

$$\begin{aligned}
&\left(\overline{V}_h^t(x_h^t) - V_h^{*,\nu^t}(x_h^t)\right) - \left(\overline{V}_{h+1}^t(x_{h+1}^t) - V_{h+1}^{*,\nu^t}(x_{h+1}^t)\right) \\
=& \underbrace{\left(\overline{V}_h^t(x_h^t) - V_h^{*,\nu^t}(x_h^t)\right) - \left(\overline{Q}_h^t(x_h^t, a_h^t, b_h^t) - Q_h^{*,\nu}(x_h^t, a_h^t, b_h^t)\right)}_{\mathrm{I}_1} \\
&+ \underbrace{\left(\overline{Q}_h^t(x_h^t, a_h^t, b_h^t) - Q_h^{*,\nu}(x_h^t, a_h^t, b_h^t)\right) - \left(\mathbb{P}_h(\overline{V}_{h+1}^t - V_{h+1}^{*,\nu^t})\right)(x_h^t, a_h^t, b_h^t)}_{\mathrm{I}_2} \\
&+ \underbrace{\left(\mathbb{P}_h(\overline{V}_{h+1}^t - V_{h+1}^{*,\nu^t})\right)(x_h^t, a_h^t, b_h^t) - \left(\overline{V}_{h+1}^t(x_{h+1}^t) - V_{h+1}^{*,\nu}(x_{h+1}^t)\right)}_{\mathrm{I}_3}.
\end{aligned} \tag{5.7}$$

We denote the corresponding terms in III as I_1', I_2', I_3'. Utilizing the bound in Lemma 2, we can bound $|\mathrm{I}_2|$ and $|\mathrm{I}_3|$ by $2\beta b_h^t(z_h^t)$ via similar techniques as in the Proof of Lemma 4. For I_3 and I_3', we note that both terms are stochastic noises with mean zero. We denote them as $\alpha_{h,t}^1$ and $\alpha_{h,t}^2$ respectively.

Finally for bounding I_1 and I_1', we prove the following lemma. (For simplicity we only prove the bound for I_1, it is trivial to generalize to the bound for I_1)

Lemma 5. We have

$$\underline{V}_h^t(x_h^t) - V_h^{\pi^t,*}(x_h^t) \leq \left(\underline{Q}_h^t(x_h^t, a_h^t, b_h^t) - Q_h^{\pi^t,*}(x_h^t, a_h^t, b_h^t)\right) + \zeta_{h,t}^2 + 2\epsilon,$$

where

$$\zeta_{h,t}^2 := \mathbb{E}_{a\sim\pi_h^t(x_h^t), b\sim br(\pi_h^t(x_h^t))}\left(\underline{Q}_h^t(x_h^t, a_h^t, b_h^t) - Q_h^{\pi^t,*}(x_h^t, a_h^t, b_h^t)\right) - \left(\underline{Q}_h^t(x_h^t, a_h^t, b_h^t) - Q_h^{\pi^t,*}(x_h^t, a_h^t, b_h^t)\right).$$

Together with Eq. (5.7) we have

$$\begin{aligned}
&\left(V_h^{*,\nu^t}(x_h^t) - \overline{V}_h^t(x_h^t)\right) - \left(V_{h+1}^{*,\nu^t}(x_{h+1}^t) - \overline{V}_{h+1}^t(x_{h+1}^t)\right) \\
&+ \left(\underline{V}_h^t(x_h^t) - V_h^{\pi^t,*}(x_h^t)\right) - \left(\underline{V}_{h+1}^t(x_{h+1}^t) - V_{h+1}^{\pi^t,*}(x_{h+1}^t)\right) \\
\leq& 2\beta\overline{b}_h^t(z_h^t) + 2\beta\underline{b}_h^t(z_h^t) + \alpha_{h,t}^1 + \alpha_{h,t}^2 + \zeta_{h,t}^1 + \zeta_{h,t}^2 + 4\epsilon.
\end{aligned} \tag{5.8}$$

Combining Eq. (5.1), and the rest of the proof follows by bounding $\overline{b}_h^t(z_h^t), \underline{b}_h^t(z_h^t)$ and the martingale difference sequences. This concludes the whole (sketch of) proof.

REFERENCES

Abbasi-Yadkori, Y., Pál, D., & Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems* (pp. 2312–2320). 2

Ayoub, A., Jia, Z., Szepesvari, C., Wang, M., & Yang, L. F. (2020). Model-based reinforcement learning with value-targeted regression. *arXiv preprint arXiv:2006.01107*. 4, 5

Bai, Y. & Jin, C. (2020). Provable self-play algorithms for competitive reinforcement learning. In *International Conference on Machine Learning* (pp. 551–560).: PMLR. 1

Bai, Y., Jin, C., & Yu, T. (2020). Near-optimal reinforcement learning with self-play. *Advances in Neural Information Processing Systems*, 33. 1

Brown, N. & Sandholm, T. (2019). Superhuman AI for multiplayer poker. *Science*, 365(6456), 885–890. 1

Chen, Z., Zhou, D., & Gu, Q. (2021). Almost optimal algorithms for two-player markov games with linear function approximation. *arXiv preprint arXiv:2102.07404*. 1, 2, 4

Chowdhury, S. R. & Gopalan, A. (2017). On kernelized multi-armed bandits. In *International Conference on Machine Learning* (pp. 844–853).: PMLR. 5

Cui, Q. & Yang, L. F. (2020). Minimax sample complexity for turn-based stochastic game. *arXiv preprint arXiv:2011.14267*. 1

Daskalakis, C., Goldberg, P. W., & Papadimitriou, C. H. (2009). The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, 39(1), 195–259. 6

Gao, R., Cai, T., Li, H., Hsieh, C.-J., Wang, L., & Lee, J. D. (2019). Convergence of adversarial training in overparametrized neural networks. *Advances in Neural Information Processing Systems*, 32, 13029–13040. 13

Huang, B., Lee, J. D., Wang, Z., & Yang, Z. (2021). Towards general function approximation in zero-sum markov games. *arXiv preprint arXiv:2107.14702*. 3

Jia, Z., Yang, L., Szepesvari, C., & Wang, M. (2020). Model-based reinforcement learning with value-targeted regression. In *Learning for Dynamics and Control* (pp. 666–686).: PMLR. 2, 4, 5

Jia, Z., Yang, L. F., & Wang, M. (2019). Feature-based Q-learning for two-player stochastic games. *arXiv preprint arXiv:1906.00423*. 1

Jiang, N., Krishnamurthy, A., Agarwal, A., Langford, J., & Schapire, R. E. (2017). Contextual decision processes with low bellman rank are pac-learnable. In *International Conference on Machine Learning* (pp. 1704–1713).: PMLR. 2

Jin, C., Liu, Q., & Yu, T. (2021). The power of exploiter: Provable multi-agent rl in large state spaces. *arXiv preprint arXiv:2106.03352*. 1, 2, 3

Jin, C., Yang, Z., Wang, Z., & Jordan, M. I. (2020). Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory* (pp. 2137–2143). 2

Lagoudakis, M. G. & Parr, R. (2002). Value function approximation in zero-sum markov games. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence* (pp. 283–292).: Morgan Kaufmann Publishers Inc. 1

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994* (pp. 157–163). Elsevier. 1

Liu, Q., Yu, T., Bai, Y., & Jin, C. (2020). A sharp analysis of model-based reinforcement learning with self-play. *arXiv preprint arXiv:2010.01604*. 1

Modi, A., Jiang, N., Tewari, A., & Singh, S. (2020). Sample complexity of reinforcement learning using linearly combined model ensembles. In *International Conference on Artificial Intelligence and Statistics* (pp. 2010–2020).: PMLR. 2

Osband, I. & Van Roy, B. (2014). Model-based reinforcement learning and the eluder dimension. *arXiv preprint arXiv:1406.1853*. 2

Pérolat, J., Piot, B., Geist, M., Scherrer, B., & Pietquin, O. (2016a). Softened approximate policy iteration for Markov games. In *International Conference on Machine Learning* (pp. 1860–1868).: PMLR. 1

Perolat, J., Piot, B., & Pietquin, O. (2018). Actor-critic fictitious play in simultaneous move multistage games. In *International Conference on Artificial Intelligence and Statistics* (pp. 919–928). 3

Pérolat, J., Piot, B., Scherrer, B., & Pietquin, O. (2016b). On the use of non-stationary strategies for solving two-player zero-sum markov games. In *Artificial Intelligence and Statistics* (pp. 893–901). 1

Perolat, J., Scherrer, B., Piot, B., & Pietquin, O. (2015). Approximate dynamic programming for two-player zero-sum Markov games. In *International Conference on Machine Learning* (pp. 1321–1329). 1

Pérolat, J., Strub, F., Piot, B., & Pietquin, O. (2017). Learning nash equilibrium for general-sum markov games from batch data. In *Artificial Intelligence and Statistics* (pp. 232–241).: PMLR. 1

Russo, D. & Van Roy, B. (2013). Eluder dimension and the sample complexity of optimistic exploration. In *NIPS* (pp. 2256–2264).: Citeseer. 2

Shalev-Shwartz, S., Shammah, S., & Shashua, A. (2016). Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*. 1

Shapley, L. S. (1953). Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10), 1095–1100. 1

Sidford, A., Wang, M., Yang, L., & Ye, Y. (2020). Solving discounted stochastic two-player games with near-optimal time and sample complexity. In *International Conference on Artificial Intelligence and Statistics* (pp. 2992–3002).: PMLR. 1

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. 1

Srinivas, N., Krause, A., Kakade, S. M., & Seeger, M. (2009). Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*. 6

Sun, W., Jiang, N., Krishnamurthy, A., Agarwal, A., & Langford, J. (2019). Model-based RL in contextual decision processes: PAC bounds and exponential improvements over model-free approaches. In *Conference on Learning Theory* (pp. 2898–2933).: PMLR. 2

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. 1

Wang, R., Salakhutdinov, R. R., & Yang, L. (2020). Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. *Advances in Neural Information Processing Systems*, 33. 2

Wei, C.-Y., Hong, Y.-T., & Lu, C.-J. (2017). Online reinforcement learning in stochastic games. In *Advances in Neural Information Processing Systems* (pp. 4987–4997). 1, 3

Xie, Q., Chen, Y., Wang, Z., & Yang, Z. (2020). Learning zero-sum simultaneous-move markov games using function approximation and correlated equilibrium. In *Conference on Learning Theory, to appear*. 1, 2, 3, 5, 6

Yang, L. & Wang, M. (2020). Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning* (pp. 10746–10756).: PMLR. 2

Yang, Z., Jin, C., Wang, Z., Wang, M., & Jordan, M. I. (2020). On function approximation in reinforcement learning: Optimism in the face of large state spaces. *arXiv preprint arXiv:2011.04622*. 2, 5, 13, 14

Zanette, A., Lazaric, A., Kochenderfer, M., & Brunskill, E. (2020). Learning near optimal policies with low inherent bellman error. In *International Conference on Machine Learning* (pp. 10978–10989).: PMLR. 2

Zhou, D., Gu, Q., & Szepesvari, C. (2021). Nearly minimax optimal reinforcement learning for linear mixture markov decision processes. In *Conference on Learning Theory*: PMLR. 2, 4

Zhou, D., He, J., & Gu, Q. (2020a). Provably efficient reinforcement learning for discounted MDPs with feature mapping. *arXiv preprint arXiv:2006.13165*. 2, 7

Zhou, D., Li, L., & Gu, Q. (2020b). Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning* (pp. 11492–11502).: PMLR. 5