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

Chris Junchi Li[◇,*] **Dongruo Zhou**[‡,*] **Quanquan Gu**[‡] **Michael I. Jordan**[◇,†]

Department of Electrical Engineering and Computer Sciences, University of California, Berkeley[◇]
Department of Statistics, University of California, Berkeley[†]
Department of Computer Sciences, University of California, Los Angeles[‡]

ABSTRACT

We consider learning Nash equilibrium in two-player zero-sum Markov games with nonlinear function approximation, where the action-value function is approximated by a function in the Reproducing Kernel Hilbert Space (RKHS). The key challenge is how to do exploration in the high-dimensional function space. We propose novel online learning algorithms to find an approximate 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 the first complexity results for learning two-player zero-sum Markov games with nonlinear function approximation in the mixture model 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 guarantees that depend 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 recent successful 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) and Chen et al. (2021) studied RL for two-player zero-sum 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., kernel and 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*,

[*] Equal contribution.

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 and statistically efficient RL algorithm for learning two-player zero-sum Markov Games with nonlinear function approximation?*

In this paper, we give an affirmative answer to this question for a class of episodic Markov Games, dubbed *mixture Markov Games*, when using nonlinear approximation function in the Reproducing Kernel Hilbert Space (RKHS). We propose a novel kernel-based MARL algorithmic framework for general episodic two-player MGs, which provides provable regret guarantees. We summarize the contributions of our work as follows:

- We propose a specific `KernelCCE-VTR` algorithm for two-player zero-sum MGs. In particular, at each episode, `KernelCCE-VTR` uses kernel function approximation 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 select an action based on the current state. In contrast to algorithms in Jin et al. (2021), which construct implicit confidence sets that are in general computationally intractable, our algorithm `KernelCCE-VTR` crafts a computationally efficient exploration bonus based on the gram matrix of the kernel function.
- Under the assumption that the transition dynamics belongs to some RKHS, we show that our algorithm `KernelCCE-VTR` is able to find an approximate Nash equilibrium of the game with a $\widetilde{O}(d_{\mathcal{F}}H^2\sqrt{T})$ upper bound on the regret of the duality gap, where H is the horizon, T is the number of the episodes, and $d_{\mathcal{F}}$ represents the complexity of the function class $\mathcal{F}$. When $\mathcal{F}$ reduces to the d-dimensional 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.
- We also study the general case where the transition dynamic belongs to some RKHS up to a misspecification error. We show that our `KernelCCE-VTR` can achieve a similar regret as in the well-specified case. In particular, we study the neural network function approximation case which can be regarded as a special instance of the misspecified RKHS case and derive the corresponding regret bound.

Notation 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 real t and interval $[a, b]$, we use $\Pi_{[a,b]}[t]$ to indicate the projection of t onto $[a, b]$, i.e. $\Pi_{[a,b]}[t] = \max(a, \min(b, t))$. For positive integer N we sometimes define $[N] = \{1, \ldots, N\}$ for compactness. We also adopt the standard big-O and big-Ω notations: 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 notations $\widetilde{O}$ and $\widetilde{\Omega}$ are adopted when the C above hides a polylogarithmic factor.

2 Related Work

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 work studying RL with function approximation. The first line of work studies the so-called linear MDP which assumes the reward function and transition dynamics are linear functions of a feature mapping defined on the state and action spaces (Yang & Wang, 2020; Jin et al., 2020; Zanette et al., 2020). These works proposed model-free algorithms with sublinear regret on the number of episodes T. The second line of work studies 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 studies proposed model-based RL algorithms that estimate the transition kernel with finite sample complexity or sublinear regret guarantees. The third line of work studies general function approximation which assumes that 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*. For example, for linear function approximation, Xie et al. (2020) studied zero-sum simultaneous-move 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 $\widetilde{O}(\sqrt{d^3H^4T})$ regret, where d is the number of the feature dimension, H is the episode length and T is the number of episodes. Chen et al. (2021) studied the linear mixture MGs and proposed a nearly minimax optimal Nash-UCRL-VTR algorithm with an $\widetilde{O}(d\sqrt{H^3T})$ regret and an $\Omega(d\sqrt{H^3T})$ matching lower bound. In contrast to this work, 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 $\widetilde{O}(H\sqrt{dT\log N})$ regret, where d is the multi-agent Bellman eluder dimension. Huang et al. (2021) studied two-player MGs with a finite minimax Eluder dimension and proposed an ONEMG method with an $\widetilde{O}(H\sqrt{dT\log N})$ 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, which is computationally intractable even in the linear function approximation setting. In contrast to Jin et al. (2021); Huang et al. (2021), our proposed `KernelCCE-VTR` is computationally efficient.

After the initial submission of this paper, we become aware of a recent independent work (Qiu et al., 2021) which also studied the kernel function approximation for two-player MGs. Here we highlight the differences between these two works. First, Qiu et al. (2021) studied the MGs where the expectation of the value function can be parameterized by a function in some RKHS, while we assume the transition dynamic of the MGs lies in an RKHS. Second, while the regret result in Qiu et al. (2021) depends on the covering number of the function space, our regret is *independent* of the covering number.

3 Preliminaries

In this section, we present the definition of a two-player Markov Game. We define the action value function corresponding to two players' 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 Coarse Correlated Equilibrium (CCE), which serves as a computational efficient approximation of the NE. Lastly, we define a mixture Markov Game setting in which the transition probability lies in an RKHS. In our context, as a generalization of a fixed kernel function, we define a weighted kernel function that depends on a pair of bounded value functions and serves as the basis of the algorithm in Section 4.

3.1 Two-Player Zero-Sum Markov Games

In this subsection, we describe simultaneous-move games in the setting of two-player Markov Games (MG). In the rest of this paper by "game" we mean "zero-sum game" unless otherwise specified. A simpler instance of Markov Games, referred to as turn-based games, can be seen as a special case of simultaneous-move games.

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 $[-1, 1]$. In the zero-sum setting, the positive reward for the max-player is the negative reward for the min-player. 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 notation we let $\mathcal{A}_1 = \mathcal{A}_2 = \mathcal{A}$, 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 Wei et al. (2017); Perolat et al. (2018); Xie et al. (2020).

In the above setting, 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_h 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 $\nu_h(\cdot \mid x_h)$:

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

and $V^{\pi,\nu}(x) := V_1^{\pi,\nu}(x)$. Note that the expectation is taken over all stochasticity in π_h, ν_h and $\mathbb{P}_h$. The action value function is defined as

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

and $Q^{\pi,\nu}(x, a, b) := Q_1^{\pi,\nu}$. From the definition, we observe that for $\forall x \in \mathcal{S}$, the state value function given policy pair (π, ν) is the expectation of the corresponding action value function

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

where the expectation is taken over the action distribution induced by the policy pair.

3.2 Equilibrium and Duality Gap

In this subsection, we recap the concepts of equilibrium and duality gap that have been widely used in the game theory literature.

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 $\mathrm{br}(\nu)$ and define $V_h^{*,\nu} = V_h^{\mathrm{br}(\nu),\nu} := \max_\pi V_h^{\pi,\nu}$ and $Q_h^{*,\nu} = Q_h^{\mathrm{br}(\nu),\nu} = \max_\pi Q_h^{\pi,\nu}$. Similarly we define $V_h^{\pi,*} := \min_\nu V_h^{\pi,\nu}$ and $Q_h^{\pi,*} := \min_\nu Q_h^{\pi,\nu}$. The Nash Equilibrium (NE) 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

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

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 call it the *Regret* in the MG setting.

Coarse Correlated Equilibrium We introduce the *Coarse Correlated Equilibrium (CCE)* (Aumann, 1987; Moulin & Vial, 1978) first. 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 for $\forall a' \in \mathcal{A}, b' \in \mathcal{B}$:

$$\mathbb{E}_{(a,b)\sim\sigma}\left[Q_1(x, a, b)\right] \ge \mathbb{E}_{b\sim\mathcal{P}_2\sigma}\left[Q_1(x, a', b)\right], \qquad \mathbb{E}_{(a,b)\sim\sigma}\left[Q_2(x, a, b)\right] \le \mathbb{E}_{a\sim\mathcal{P}_1\sigma}\left[Q_2(x, a, b')\right],$$

where $\mathcal{P}_1\sigma$ denotes the marginal of σ on the first coordinate (min-player) and $\mathcal{P}_2\sigma$ denotes the marginal of σ on the second coordinate (max-player). As suggested by Xie et al. (2020), the set of CCE includes the set of Correlated Equilibrium (CE) (Aumann, 1987; Moulin & Vial, 1978; Blum & Monsour, 2007), which further includes the set of NE. Even though the set of CCE is known as a convex set (Osborne & Rubinstein, 1994), later we will show that a good approximation of the NE can be derived from the approximate CCE for the estimates of the value function corresponding to P_1 and P_2, respectively.

3.3 Nonlinear Function Approximation by Reproducing Kernel Hilbert Spaces

In this subsection, we provide necessary definitions and notations in approximating action value function with functions belonging to an reproducing kernel Hilbert space (RKHS) via modeling the transition probability. For the simplicity of notation, 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 $\mathbb{P}_h(s' \mid z) : \mathcal{Z} \mapsto \Delta(\mathcal{S})$ as an element in an RKHS $\mathcal{H}$ with feature mapping $\phi(s' \mid z) : \mathcal{Z} \to \mathcal{S} \times \mathcal{H}$, such that the following equality holds for 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 element in the RKHS

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

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 mapping with the estimated value of s' over $\mathcal{S}$. It is worth noting that the quantity $\phi_V(\cdot)$ plays an important role in previous linear mixture model-based algorithms (Jia et al., 2020; Ayoub et al., 2020; Zhou et al., 2021; Chen et al., 2021).

In this work, we face a general RKHS $\mathcal{H}$ and we cannot access the feature mapping ϕ directly. Instead, we assume that we can access the *weighted kernel function* $k_{V_1,V_2}(\cdot, \cdot)$, which is defined as follows:

Definition 1. For any function pairs $V_1, V_2 : \mathcal{S} \to [0, 1]$ which map states to real numbers, the weighted kernel function $k_{V_1,V_2}(\cdot, \cdot)$ is defined as follows for all $z_1, z_2 \in \mathcal{Z}$.

$$k_{V_1,V_2}(z_1, z_2) := \sum_{s_1, s_2 \in \mathcal{S}} V_1(s_1) V_2(s_2) \langle \phi(s_1|z_1), \phi(s_2|z_2) \rangle_{\mathcal{H}},$$

It is easy to see that

$$k_{V_1,V_2}(z_1, z_2) = \left\langle \sum_{s_1 \in \mathcal{S}} V_1(s_1)\phi(s_1|z_1), \sum_{s_2 \in \mathcal{S}} V_2(s_2)\phi(s_2|z_2) \right\rangle_{\mathcal{H}} = \langle \phi_{V_1}(z_1), \phi_{V_2}(z_2) \rangle_{\mathcal{H}},$$

which suggests that the weighted kernel function $k_{V_1,V_2}(\cdot, \cdot)$ indeed captures the inner product relation between $\phi_{V_1}(z_1)$ and $\phi_{V_2}(z_2)$. In 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 $z \in \mathcal{Z}$, $\|\phi_V(z)\|_{\mathcal{H}} \leq 1$. Given that the reward function $r_h(z)$ is known, we obtain through the Bellman equation that

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

and similar equation holds for $Q_h^{\pi,*}$. With these preliminaries 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 zero-sum Markov Game setting with 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), and combine the CCE technique (Xie et al., 2020) to deal with the general sum sub-game brought by upper confidence bound (UCB) and lower confidence bound (LCB) value functions. These techniques enable us to adapt the results in the linear setting to the nonlinear RKHS regime (Chowdhury & Gopalan, 2017; Yang et al., 2020; Zhou et al., 2020b) 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 confidence bound estimations. As the min-player targets the minimization of the value function while the max-player targets the maximization of the value function, we use upper confidence bound estimation to encourage exploration of the max-player and use a lower confidence bound to encourage exploration of the min-player. Thus we need to define two value functions for the min/max-players respectively, i.e., $\overline{Q}_h^t, \underline{Q}_h^t, \overline{V}_h^t, \underline{V}_h^t$, 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 the game, we solve the following ridge regression problem for minimizing the Bellman error:

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

Note that in Eq. (4.1), $\overline{V}_{h+1}^{\tau}, \underline{V}_{h+1}^{\tau}$ only depend 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 $\overline{V}_{h+1}^{\tau}(x_{h+1}^{\tau})$ 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 solving a linear 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 $\overline{V}_{h+1}^{\tau}(x_{h+1}^{\tau}) - (\mathbb{P}_h\overline{V}_{h+1}^{\tau})(z_h^{\tau})$. A similar statement holds for the min-player as well.

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

$$\overline{\boldsymbol{\Psi}}_h^t := \left(\phi_{\overline{V}_{h+1}^1}(z_h^1), \ldots \phi_{\overline{V}_{h+1}^{t-1}}(z_h^{t-1})\right)^{\top}, \quad \text{and} \quad \underline{\boldsymbol{\Psi}}_h^t := \left(\phi_{\underline{V}_{h+1}^1}(z_h^1), \ldots \phi_{\underline{V}_{h+1}^{t-1}}(z_h^{t-1})\right)^{\top}.$$

Also, we define the gram matrix $\overline{K}_h^t$ and vector-valued function $\overline{k}_h^t$ as

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

separately. 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 $\overline{\boldsymbol{\theta}}_h^t$ in the RKHS $\mathcal{H}$ is defined as

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

where we use $\left\langle \overline{\boldsymbol{\Psi}}_h^t, \boldsymbol{\theta} - \overline{\boldsymbol{\theta}}_h^t\right\rangle_{\mathcal{H}}$ to denote the following $t-1$ dimensional vector $\left(\left\langle \phi_{\overline{V}_{h+1}^{\tau}}(z_h^{\tau}), \boldsymbol{\theta} - \overline{\boldsymbol{\theta}}_h^t\right\rangle_{\mathcal{H}} : \tau \in [t-1]\right)$. 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 ones. Based on the confidence regions, we construct an optimistic/pessimistic estimate of $Q_h^{*,\nu}$ as

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

where $\Pi_{[-H,H]}$ is the projection operator onto $[-H, H]$, which is by definition the range of value functions. In fact, $\overline{Q}_h^t$ has a closed-form solution as follows:

$$\overline{Q}_h^t(z) = \Pi_{[-H,H]}\left[r_h(z) + \overline{k}_h^t(z)^{\top}(\overline{K}_h^t + \lambda I)^{-1}\overline{y}_h^t + \beta_t \cdot \overline{b}_h^t(z)\right], \tag{4.2}$$

where $\overline{y}_h^t := \left[\overline{V}_{h+1}^1(x_h^1), \ldots \overline{V}_{h+1}^{t-1}(x_h^{t-1})\right]^{\top}$ and $\overline{b}_h^t(z) = \lambda^{-1/2} \cdot \left[k_{\overline{V}_{h+1}^t, \overline{V}_{h+1}^t}(z, z) - \overline{k}_h^t(z)^{\top}\left(\overline{K}_h^t + \lambda \cdot \mathbf{I}\right)^{-1}\overline{k}_h^t(z)\right]^{1/2}$. An analogous claim 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 instead find a CCE of the payoff pair $(\overline{Q}_h^t(z), \underline{Q}_h^t(z))$. More specifically, we utilize the `FIND_CCE` algorithm in Xie et al. (2020). The full version of the algorithm is presented formally in Algorithm 1.

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 $h = 1, 2, \ldots, H$ **do**
    Sample $a_h^t \sim \pi_h^t(x_h^t) := \mathcal{P}_1\sigma_h^t(x_h^t)$, $b_h^t \sim \nu_h^t(x_h^t) := \mathcal{P}_2\sigma_h^t(x_h^t)$.
    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

5 Theoretical Analysis

In this section, we present the regret bound of our algorithm for the kernel mixture Markov Game. Recall that for the 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 (Chen et al., 2021). Our analysis in the RKHS function approximation setting aligns with the linear function approximation setting when $k_{V,V'}(z,z') = \phi_V(z)^\top \phi_{V'}(z')$ and yields a regret bound of $\widetilde{\mathcal{O}}(dH^2\sqrt{T})$.

When considering the nonlinear function class as an approximator of the value function, we need to develop a new concept analogous to the dimension d that characterizes 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 lay out precise definitions of these notions immediately afterwards.

We first define the *effective dimension* of the RKHS $\mathcal{H}$ with respect to the mixture MG as follows:

Definition 2 (Srinivas et al. 2009)**.** We define the effective dimension $\Gamma_K(T,\lambda)$ as follows:

$$\Gamma_K(T,\lambda) := \sup_{(V_i)_i,(z_i)_i} \frac{1}{2}\log\det(\mathbf{I} + K(\{V_i\}_i,\{z_i\}_i)/\lambda),$$

for any $1 \le i \le T$, $V_i : \mathcal{S} \to [-H,H]$, $z_i \in \mathcal{Z}$, where V_i's are functions mapping from $\mathcal{S}$ to $[-H,H]$ and z_i's are state-action tuples. Here, $K(\{V_i\}_i,\{z_i\}_i) \in \mathbb{R}^{T\times T}$ and its (p,q)-th entry for any $1 \le p,q \le T$ is $[K(\{V_i\}_i,\{z_i\}_i)]_{p,q} = k_{V_p,V_q}(z_p,z_q)$.

By the boundedness of ϕ_V as in Section 3.3, it is easy to verify that both the tabular MG and the linear mixture MG enjoy a finite effective dimension. Specifically, for finite RKHS $\mathcal{H}$ with rank d, $\Gamma_K(T,\lambda) = \mathcal{O}(d\cdot\log T)$ approximately depicts the rank of $\mathcal{H}$. Via a concentration argument, we first present our main lemma for bounding the estimation error when choosing $\beta_t = \beta$ for all $t \ge 1$:

Lemma 3. 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 $(x,a,b) \in \mathcal{S}\times\mathcal{A}\times\mathcal{A}$:

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

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

Theorem 4 (RKHS function approximation)**.** 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

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

there exists a universal constant $c > 0$ such that with probability at least $1 - \delta$, we have

$$\text{Regret}(T) \le c\left(\beta H\sqrt{T \cdot \Gamma_K(T,\lambda)}\right).$$

Remark 5. Theorem 4 suggests that by treating the norm B as a constant, `KernelCCE-VTR` achieves an $\widetilde{O}(\Gamma_K(T,\lambda)H^2\sqrt{T})$ regret bound. When the RKHS degenerates to the Euclidean case, the regret bound corresponding to the standard linear mixture MG case reduces to $\widetilde{O}(dH^2\sqrt{T})$ which matches the $\widetilde{O}(dH^{3/2}\sqrt{T})$ regret yielded by Chen et al. (2021) up to a $\sqrt{H}$ factor. The $\widetilde{O}(dH^2\sqrt{T})$ regret is also known to be nearly minimax optimal up to a $\sqrt{H}$ factor (Chen et al., 2021).

Remark 6. Theorem 4 suggests that the average of the duality gaps $1/T \cdot \sum_{t=1}^{T}(V_1^{*,\nu^t} - V_1^{\pi^t,*}) = 1/T \cdot \text{Regret}(T) \to 0$, which further indicates that our `KernelCCE-VTR` algorithm can indeed find a good approximation of the Nash Equilibrium (π_h^t, ν_h^t), as the marginalization of the CCE σ_h^t.

6 Nonlinear Function Approximation with Misspecification

In Section 5, we focus on the case where the transition probability $\mathbb{P}_h(s' \mid z)$ can be modeled by functions belonging to an RKHS. In practice, however, the function class may not be confined to an RKHS, but the distance to it can be bounded. For example, when considering the more general Neural Network (NN) function classes, it is well-known that we can approximate it by the RKHS with the corresponding neural tangent kernel (Jacot et al., 2018; Allen-Zhu et al., 2019). In this section, we attempt to resolve this in two parts, the RKHS approximation with misspecification (Section 6.1), and the NN approximation (Section 6.2) as an application of the misspecification results.

6.1 Kernel Function Approximation with Misspecification

In this subsection, we discuss the case where there exists a misspecification error between the RKHS $\mathcal{H}$ and the true transition probability $\mathbb{P}_h(s'|z)$. Formally, we make the following assumptions:

Assumption 7. We assume that there exists an $\iota_{\text{mis}} > 0$, an RKHS $\mathcal{H}$ with feature mapping $\phi : \mathcal{Z} \mapsto \mathcal{S} \times \mathcal{H}$, and an unknown parameter $\boldsymbol{\theta}_h^* \in \mathcal{H}$ satisfying $\|\boldsymbol{\theta}_h^*\|_{\mathcal{H}} \le B$ such that for any $h \in [H]$, the distance of the transition probability $\mathbb{P}_h$ to $\mathcal{H}$ can be bounded by ι_{mis}:

$$\|\mathbb{P}_h(\cdot \mid z) - \langle \phi(\cdot \mid z), \boldsymbol{\theta}_h^* \rangle_{\mathcal{H}}\|_{\text{TV}} \le \iota_{\text{mis}}.$$

Similar to Lemma 3 in Section 5, by choosing a proper size β_t of the confidence region, we are able to bound the estimation error:

Lemma 8. Assuming that for any $h \in [H]$, $||\boldsymbol{\theta}_h^*||_{\mathcal{H}} \le B$. Let $\lambda = 1 + \frac{1}{T}$ and β_t satisfies

$$\left(\frac{\beta_t}{H}\right)^2 \ge 3\Gamma_K(T,\lambda) + 3 + 6 \cdot \log\left(\frac{1}{\delta}\right) + 3\lambda\left(\frac{B}{H}\right)^2 + 3\iota_{\text{mis}}^2 t. \tag{6.1}$$

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

$$\left|\left\langle \phi_{\overline{V}_{h+1}^t}(z), \overline{\boldsymbol{\theta}}_h^t \right\rangle_{\mathcal{H}} - \mathbb{P}_h \overline{V}_{h+1}^t(z)\right| \le \beta_t \cdot \overline{b}_h^t(z) + H \cdot \iota_{\text{mis}},$$
$$\left|\left\langle \phi_{\underline{V}_{h+1}^t}(z), \underline{\boldsymbol{\theta}}_h^t \right\rangle_{\mathcal{H}} - \mathbb{P}_h \underline{V}_{h+1}^t(z)\right| \le \beta_t \cdot \underline{b}_h^t(z) + H \cdot \iota_{\text{mis}}.$$

Compared with the choice of β in Lemma 3, Eq. (6.1) yields an extra $\mathcal{O}(H\iota_{\text{mis}}\sqrt{t})$ term brought by misspecification error. Now we are ready to present the regret bound when the misspecification occurs.

Theorem 9 (RKHS function approximation with misspecification)**.** 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_t satisfying

$$\left(\frac{\beta_t}{H}\right)^2 \ge 2\Gamma_K(T,\lambda) + 3 + 6 \cdot \log\left(\frac{1}{\delta}\right) + 3\lambda\left(\frac{B}{H}\right)^2 + 3\iota_{\text{mis}}^2 t,$$

there exists a global constant $c > 0$ such that with probability at least $1 - \delta$, we have

$$\text{Regret}(T) \le c\left(\beta_T H\sqrt{T \cdot \Gamma_K(T,\lambda)} + H^2 T \iota_{\text{mis}}\right).$$

In words, Theorem 9 suggests that in the misspeficified case, `KernelCCE-VTR` can achieve the same regret as that in the well-specified case up to an $O(H^2 T \iota_{\text{mis}})$ error. Such a linear dependence on ι_{mis} matches the result of single agent RL for the finite dimensional case (Jin et al., 2020; Zanette et al., 2020).

6.2 Neural Network (NN) Function Approximation

The previous subsection focuses on estimating the transition probability using an RKHS when misspecification is present. In this subsection, we utilize Theorem 9 in deriving a regret bound for neural network function approximations. We see $w := (x', x, a, b)^\top = (x', z)$ as a vector in $\mathbb{R}^d$ that satisfies $||w|| = 1$ and represent the parameters of a L-Layer fully connected neural network f by $\boldsymbol{\theta} := \left[\text{vec}(\boldsymbol{W}_1)^\top, \text{vec}(\boldsymbol{W}_2)^\top, \ldots, \text{vec}(\boldsymbol{W}_L)^\top\right]^\top$, where $\boldsymbol{W}_i \in \mathbb{R}^{m \times m}$ for all $i \in [L]$. The neural network $f(w; \boldsymbol{\theta})$ with parameter set $\boldsymbol{\theta}$ can be defined as:

$$f(w; \boldsymbol{\theta}) = \sqrt{m} \boldsymbol{W}_L G\left(\cdots G\left(\boldsymbol{W}_2 G\left(\boldsymbol{W}_1 w\right)\right)\right),$$

where $G(\cdot) : \mathbb{R} \mapsto \mathbb{R}$ is an activation function, $\boldsymbol{W}_L \in \mathbb{R}^{m \times 1}$, $\boldsymbol{W}_l \in \mathbb{R}^{m \times m}, 2 \leq l < L$, $\boldsymbol{W}_1 \in \mathbb{R}^{m \times d}$. For $1 \leq l \leq L-1$, $\boldsymbol{W}_l = (\boldsymbol{W}, \boldsymbol{0}; \boldsymbol{0}, \boldsymbol{W})$, where each entry of $\boldsymbol{W}$ is generated independently from $N(0, 4/m)$; $\boldsymbol{W}_L = (\boldsymbol{w}^\top, -\boldsymbol{w}^\top)$ where each entry of $\boldsymbol{w}$ is generated independently from $N(0, 2/m)$. Given the initialized parameter $\boldsymbol{\theta}_0$, we take the feature map $\phi(w) = \nabla_{\boldsymbol{\theta}} f(w; \boldsymbol{\theta}_0)/\sqrt{m}$ as the gradient of f at $\boldsymbol{\theta}_0$. We define the weighted kernel function $k_{V_1, V_2}(\cdot, \cdot)$ in Definition 1 with $\phi(w)$. Similarily, we define the effective dimension $\Gamma_K(T, \lambda)$ with respect to the kernel function $k_{V_1, V_2}(\cdot, \cdot)$, in the same fashion of Definition 2.

We assume that for $\forall h \in [H]$ our transition probability $\mathbb{P}_h$ can be modeled by the neural network with parameter $\boldsymbol{\theta}_h^*$ satisfying $\left\|\boldsymbol{\theta}_h^* - \boldsymbol{\theta}^{(0)}\right\|_2 \leq B/\sqrt{m}$:

$$\mathbb{P}_h(x' \mid z) = f(x', z; \boldsymbol{\theta}_h^*).$$

With these at hand we are ready to present our main theorem for NN approximation:

Theorem 10 (NN approximation)**.** Assuming that for any $h \in [H]$, $||\boldsymbol{\theta}_h^* - \boldsymbol{\theta}^{(0)}||_2 \leq B/\sqrt{m}$. There exists a $C > 0$ independent of m such that if we $\lambda = C^2\left(1 + \frac{1}{T}\right)$ in the `KernelCCE-VTR` Algorithm. For any $\delta > 0$ and any β_t satisfying

$$\left(\frac{\beta_t}{H}\right)^2 \geq 2\Gamma_K(T, \lambda) + 3 + 6 \cdot \log\left(\frac{1}{\delta}\right) + 3\lambda\left(\frac{B}{H}\right)^2 + 3 \cdot C^2 \cdot B^{8/3} \cdot m^{-1/12} \cdot t \cdot \log m,$$

there exists a global constant $c > 0$ such that with probability at least $1 - \delta - m^{-2}$, we have

$$\text{Regret}(T) \leq c\left(\beta_T H \sqrt{T \cdot \Gamma_K(T, \lambda)} + B^{4/3} H^2 T m^{-1/6} \sqrt{\log m}\right).$$

Theorem 10 suggests that when we use an overparameterized deep neural network ($m \gg 1$) to approximate the transition dynamic, `KernelCCE-VTR` achieves an $\widetilde{O}(\Gamma_K(T, \lambda) H^2 \sqrt{T})$ regret, which is of the same order as its counterparts in the general misspecification case.

Remark 11. To match the setting of the misspecification case, we consider solving the NN approximation case by a simplified version of Algorithm 1 that uses only the first-order Taylor expansion compared with the full NN solution in Zhou et al. (2020b); Yang et al. (2020).

7 Conclusions

In this work, we studied learning two-player mixture MGs using the kernel function approximation. We introduced a new kernel mixture MG setting and proposed a new algorithm `KernelCCE-VTR` that utilizes the kernel function of the MG. We show that our `KernelCCE-VTR` is able to achieve a sublinear $\widetilde{O}(d_{\mathcal{F}} H^2 \sqrt{T})$ regret, which nearly matches the regret lower bound in Zhou et al. (2021) for learning linear mixture MGs. We further extend the analysis of the basic RKHS setting to a more general nonlinear function approximation with misspecification errors at present and demonstrate that neural networks can be seen as a special application of misspecification.

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

Allen-Zhu, Z., Li, Y., & Song, Z. (2019). A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning* (pp. 242–252).: PMLR. 8

Aumann, R. J. (1987). Correlated equilibrium as an expression of Bayesian rationality. *Econometrica: Journal of the Econometric Society*, (pp. 1–18). 4

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*. 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

Blum, A. & Monsour, Y. (2007). *Learning, Regret Minimization, and Equilibria*. Cambridge University Press. 4

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, 3, 5, 7, 8

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. 14

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

Jacot, A., Gabriel, F., & Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31. 8

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, 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, 9

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

Moulin, H. & Vial, J.-P. (1978). Strategically zero-sum games: the class of games whose completely mixed equilibria cannot be improved upon. *International Journal of Game Theory*, 7(3-4), 201–221. 4

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

Osborne, M. J. & Rubinstein, A. (1994). *A course in game theory*. MIT press. 4

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

Qiu, S., Ye, J., Wang, Z., & Yang, Z. (2021). 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. 3

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*. 7

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, 3, 4, 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, 9, 14, 15

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, 9

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, 5, 9

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

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, 9